ON DISCRIMINATIVE SEMI-SUPERVISED INCREMENTAL LEARNING WITH A MULTI-VIEW PERSPECTIVE FOR IMAGE CONCEPT MODELING

A Thesis
Presented to
The Academic Faculty

by

Byungki Byun

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2012

ON DISCRIMINATIVE SEMI-SUPERVISED INCREMENTAL LEARNING WITH A MULTI-VIEW PERSPECTIVE FOR IMAGE CONCEPT MODELING

Approved by:

Professor James H. McClellan,
Committee Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Hsien-Hsin Lee
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Chin-Hui Lee, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Ming Yuan
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Mark A. Clements
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 13 January 2012

To my parents,

Yeonggil Byun and Meesook Jung,

my son, Heehoon,

and my wife, Jiyoung Lee

ACKNOWLEDGEMENTS

Reminiscing the road I traveled for this research journey, it has been one of the most fruitful experiences in my life. Every time I passed a twist or a roadblock, I gained greater insight to the true meaning of passion and the joy of achievement. The journey also provided me a great opportunity to hone my academic and professional skills. Nevertheless, having gone through the journey has exhausted a great deal of efforts. So I cannot help but feel grateful to many people around me for their guidance, encouragement, support, and love, which have rejuvernated me and kept me moving forward.

First of all, I would like to express my deepest gratitude to my advisor, Dr. Chin-Hui Lee. I am very fortunate to have him as my advisor. He has provided me invaluable advice with and generous heart throughout my tenure. Without his inspirational devotion and energy for research, I would not be able to pass through such a hazy road. His universal support of my ideas helped me keep a positive mindset whenever I encounter obstacles. Moreover, his insightful criticism enabled me to develop an appreciation of many areas, including machine learning, image processing, and speech processing.

I would also like to specifically thank Dr. James McClellan and Dr. Mark Clements for serving on the reading committee of this dissertation. Their valuable feedback on my research has provided me an opportunity to improve the quality of this work. I would also like to thank Dr. Hsien-Hsin Lee and Dr. Ming Yuan for serving on my thesis committee and for providing me with insightful comments.

I am grateful to the Center for Signal and Image Processing (CSIP) staff, Pat Dixon, Tammy Scott, and Stacie Speights. Their efforts enabled students to conduct research without problems.

I owe my special thanks to all my former and current collegues at the CSIP, Sibel Yaman, Jinyu Li, Yu Tsao, Chengyuan Ma, Jeremy Reed, You-Chi Cheng, Ifan Chen, Marco Sabato for their help and stimulating discussions throughout my journey. It has been a great

pleasure to have Bongkyung Kwon, Seonghyuk Kim, Sehun Kook, Suhwan Kim, Ilseo Kim, Sunghwan Shin, Jonathan Kim as dependable friends, with whom joyous memories have been shared.

I thank my parents, Yeonggil Byun and Meesook Jung, for their endless love, support, and sacrifice throughout my life and during my Ph.D. studies. I would also like to express thanks to my sisters, Hyunah and Hyunsook Byun, and my parents-in-law, Suha Lee and Kyeyoung Kim, for their universal support. Last but most important, my greatest gratitude goes to my wife, Jiyoung Lee, who has given me enormous support and encouragement throughout this experience. She has been so supportive and patient, with the greatest love. Without her, this work would have not been completed.

TABLE OF CONTENTS

DEDICATION . . . . . **iii**

ACKNOWLEDGEMENTS . . . . . **iv**

LIST OF TABLES . . . . . **ix**

LIST OF FIGURES . . . . . **x**

SUMMARY . . . . . **xiii**

I INTRODUCTION **1**

1.1 Organization of this dissertation 5

II BACKGROUND AND RELATED WORK **7**

2.1 Background of discriminative learning 7

2.1.1 Mathematical formulation 7

2.1.2 Existing discriminative learning techniques 9

2.1.3 Performance metrics and discriminative learning 10

2.1.4 Non-linearization of discriminative learning techniques 12

2.2 Background of discriminative semi-supervised learning (SSL) 12

2.2.1 Mathematical formulation 13

2.2.2 Methods using low-density separation 14

2.2.3 Methods using a data-driven kernel 15

2.3 Relevant incremental learning techniques 18

2.3.1 Active learning 19

2.3.2 Semi-supervised incremental learning 21

2.4 Relevant multi-view learning techniques 25

III KERNELIZED MAXIMAL-FIGURE-OF-MERIT (KMFOM) LEARNING FOR IMAGE CONCEPT MODELING **28**

3.1 Subspace distance minimization through the Nyström extension 30

3.2 The proposed kMFoM learning framework 33

3.3 Experimental results on image concept modeling 35

3.4 Summary 42

IV SEMI-SUPERVISED INCREMENTAL LEARNING WITH AN ERROR REDUCTION FUNCTION . . . 44

4.1 A confidence score . . . 46

4.2 An expected error reduction function . . . 48

4.3 Robust estimation of the expected error reduction through an ensemble of classifiers . . . 50

4.4 The proposed algorithm . . . 54

4.5 Experimental results . . . 60

4.5.1 Comparisons with other semi-supervised incremental learning algorithms . . . 62

4.5.2 Comparisons between different sizes of an ensemble . . . 68

4.6 Summary . . . 77

V AN AGREEMENT FUNCTION FOR MULTI-VIEW SEMI-SUPERVISED LEARNING (SSL) . . . 80

5.1 Algorithms to learn the agreement function . . . 83

5.1.1 Graphical representations of the disagreement noise . . . 83

5.1.2 A graph-based algorithm to learn the agreement function . . . 85

5.1.3 A probabilistic algorithm to learn the agreement function . . . 86

5.2 A multi-view SSL framework with an agreement function . . . 88

5.3 Multi-view SSL on artificial data sets . . . 90

5.4 Multi-view SSL on the TREC05 spam corpus . . . 96

5.4.1 Feature extraction techniques . . . 98

5.4.2 Experimental results . . . 101

5.5 Summary . . . 104

VI DISCRIMINATIVE SEMI-SUPERVISED INCREMENTAL LEARNING APPROACH WITH A MULTI-VIEW PERSPECTIVE . . . 106

6.1 System overview . . . 107

6.2 Experimental results . . . 109

6.3 Summary . . . 113

VII CONCLUSION AND FUTURE WORK . . . 115

7.1 Contributions of this dissertation . . . 117

7.2 Avenues for future work . . . 119

APPENDIX **120**

Bibliography **126**

LIST OF TABLES

1 The definitions of some commonly used performance metrics. For simplicity, we assume that there are only two classes; positive vs. negative. Note that precision and recalls are computed with respect to the positive class. *no.* stands for *number* 11

2 Comparisons on performances and training times while varying the size of the subset of training data 42

3 The list of chosen parameters 68

4 Comparisons of classification error rates on the TREC05 spam corpus among two single-view cases and a multi-view case 103

5 Comparisons of classification error rates on the TREC05 spam corpus between multi-view SSL frameworks with and without an agreement function 103

LIST OF FIGURES

1 An illustration of a low-density separation assumption 14

2 An example of a manifold embedded into a higher-order space 16

3 An overview of incremental learning 18

4 Two images extracted from the USPS handwritten digit recognition task data set. They correspond to two different numbers, four and nine, respectively, but their appearances are very similar to each other 29

5 Examples of handwritten digit images extracted from the USPS data set . . . 36

6 Some object images extracted from the COIL-100 data set 37

7 Some images from the Corel 5k data set and their associated semantic concepts 38

8 An illustration of the procedures to compute visual unigrams and bigrams . . 39

9 Performance comparison graphs between the proposed kernelized MFoM learning approach and the baseline system. 41

10 The values of the expected error reduction function when the size of an ensemble is two 53

11 The values of the selection scores $s(\delta(x;\boldsymbol{\theta}^t))$ when the size of an ensemble is two and $\gamma = 0.5$ 55

12 The values of the selection scores $s(\delta(x;\boldsymbol{\theta}^t))$ when the size of an ensemble is two and $\gamma = 0.42$. Lines indicate the boundaries where $s(\delta(x;\boldsymbol{\theta}^t))$ is 93.5% of its maximum value. 57

13 The number of classifiers with $P(y|x;\boldsymbol{\theta}^t) = 1$ to attain the maximum value of the selection score $s(\delta(\cdot;\boldsymbol{\theta}^t))$ against various values of γ while the number of total classifiers is varied from 2 to 16. 58

14 Performance comparison curves between the baseline systems and the proposed semi-supervised incremental learning framework for the USPS data set 65

15 Performance comparison curves between the baseline systems and the proposed semi-supervised incremental learning framework for the COIL-100 data set 67

16 Performance comparison curves on the USPS data set using 5% of training data as initially labeled samples while varying the number of classifiers in an ensemble among 2, 4, and 8 70

17 Performance comparison curves on the USPS data set using 10% of training data as initially labeled samples while varying the number of classifiers in an ensemble among 2, 4, and 8 71

18 Performance comparison curves on the USPS data set using 20% of training data as initially labeled samples while varying the number of classifiers in an ensemble among 2, 4, and 8 72

19 Performance comparison curves on the COIL data set using 5% of training data as initially labeled samples while varying the number of classifiers in an ensemble among 2, 4, and 8. 74

20 Performance comparison curves on the COIL data set using 10% of training data as initially labeled samples while varying the number of classifiers in an ensemble among 2, 4, and 8. 75

21 Performance comparison curves on the COIL data set using 20% of training data as initially labeled samples while varying the number of classifiers in an ensemble among 2, 4, and 8 76

22 Performance comparison curves on the COIL data set using 35% of training data as initially labeled samples while varying the number of classifiers in an ensemble 78

23 Simple graphical representations of possible configurations of two unlabeled samples with two views 84

24 A graphical representation of what will happen with a presence of the disagreement noise 84

25 An illustration of artificial data sets used to demonstrate the effectiveness of the agreement function in multi-view SSL 91

26 Performance comparison graphs between the baseline system and the proposed multi-view SSL technique when the number of labeled samples is set to 20 93

27 Performance comparison graphs between the baseline system and the proposed multi-view SSL technique when the number of labeled samples is set to 50 95

28 Examples of images containing spam messages 96

29 An example of an image spam email 97

30 A comparison of a text message from an image spam email with that from legitimate email 98

31 Examples of images extracted from legitimate emails 99

32 A block diagram of a discriminative semi-supervised incremental learning approach with a multi-view perspective 108

33 Performance comparison curves of the proposed semi-supervised incremental learning framework on the TREC05 spam corpus when the size of initially labeled set is 5% of the training set 112

34 Performance comparison curves of the proposed semi-supervised incremental learning framework on the TREC05 spam corpus when the size of initially labeled set is 10% of the training set 113

SUMMARY

This dissertation presents the development of a semi-supervised incremental learning framework with a multi-view perspective for image concept modeling. For reliable image concept characterization, having a large number of labeled images is crucial. However, the size of the training set is often limited due to the cost required for generating concept labels associated with objects in a large quantity of images. To address this issue, in this research we propose to incrementally incorporate unlabeled samples into a learning process to enhance concept models originally learned with a small number of labeled samples. To improve the convergence property of the proposed incremental learning framework, we further propose a multi-view learning approach that makes use of multiple features such as color, texture, etc., of images when including unlabeled samples. For robustness to mismatches between training and testing conditions, a discriminative learning algorithm, namely a kernelized maximal-figure-of-merit (kMFoM) learning approach is also developed.

A typical strategy for semi-supervised learning is to choose samples where an existing model is able to correctly predict their class labels with high confidence. However, these samples are not usually the best in terms of reducing modeling error as they are often too similar to already seen examples. In contrast, the proposed incremental learning framework selects unlabeled samples based on an *expected error reduction* function that measures contributions of the unlabeled samples based on their ability to increase the modeling accuracy. In the proposed framework, one of the essential components for robust estimation of the expected error reduction is a use of ensemble classifiers, such as a combination of a kMFoM classifiers and a spectral clustering based nearest neighbor (NN) classifier, etc. We demonstrate that, given an unlabeled example - when half of the classifiers in an ensemble predict the class label for the sample almost definitively, while the rest of them remain uncertain - the maximum value of the expected error reduction can be obtained. We generalize this result by developing iterative learning procedures that control the number of classifiers

within the ensemble that should exhibit high confidence in their classification results when selecting unlabeled samples. We further improve the stability of the proposed framework by exploiting a class prior distribution so that a potential *class imbalance problem* can be reduced.

On the other hand, taking advantage of multiple features (e.g., color, texture, etc.) is vital to achieving a good image concept model. In a semi-supervised setting, a typical method to benefit from multiple features, known as *multi-view learning*, is to enforce concept models trained on individual features to generate the same prediction result. However, such enforcement is not always beneficial because different features might preferrably indicate different class labels. Thus, in this dissertation, we propose a multi-view learning technique that exploits an agreement function, a function conveying our degree of belief that the individual models should agree upon their outputs. Then, we formulate a closed-form solution for a kernel function that represents an unified reproducing kernel Hilbert space (RKHS) with this agreement function.

Combining individual techniques, we also propose an integrated semi-supervised incremental learning framework, namely a discriminative semi-supervised incremental learning approach with a multi-view perspective, that takes advantage of both the power of multiple features and the expected error reduction function. In the integrated framework, multiple features extracted from images are combined through the kernel function. An ensemble of discriminative classifiers are then learned using the kernel function from which the expected error reduction function is computed. Based on the values of the expected error reduction, a set of unlabeled samples is chosen and exploited to enhance an existing model gradually. We conduct a set of experiments on various image concept modeling problems, such as handwritten digit recognition, object recognition, and image spam detection to highlight the effectiveness of the proposed framework.

Chapter I

INTRODUCTION

Humans learn by example [3]. It is said that a child builds a knowledge base by experiencing the surrounding world with supervision (*i.e., supervised learning*) or without supervision (*i.e., unsupervised learning*) of the child's parents or teachers [14]. Many machine learning algorithms, in fact, can be seen as mathematical modeling of such human learning processes. For example, consider one of the classical supervised machine learning problems, a point estimation problem. Here, a parameter vector is estimated by a maximum likelihood (ML) criterion (*mathematical modeling*) given a set of examples (*examples*) and their associated class labels (*supervision*). Similary, suppose we have an unsupervised learning problem such as a probability density function (PDF) estimation problem. Then, given training samples (*examples without supervision*), a PDF is estimated using a kernel density estimation (KDE) technique (*mathematical modeling*).

A fundamental assumption lying at the core of machine learning is that examples seen during training and those in testing are independently and identically distributed (i.i.d.). In many real-world applications however, this assumption is often violated as a result of mismatches between training conditions and testing conditions. Conventional ways to reduce such mismatches have been mainly two-fold: (a) to increase the size of a training data set in hopes that some of the mismatch might be mitigated by the newly included samples, and (b) to develop a learning algorithm with a small number of labeled samples that is robust to the mismatches. However, both approaches often fail because for the first approach, using more training data might not be feasible, as the amount of effort required to generate class labels is often prohibitive [76]. For the second approach, the number of training samples fundamentally limits the robustness of a learning algorithm, as studied in [99, 5].

Interestingly, a closer look at human learning processes can reveal a viable solution to such a mismatch problem. In many cases, when a child learns, the child not only relies on

instructions from parents or teachers, but he or she also examines the surrounding world without supervision *simultaneously* [119]. This observation implies that machines can also learn using labeled samples (*supervised*) as well as unlabeled samples (*unsupervised*) at the same time. Recently, this learning approach, known as *semi-supervised learning (SSL)* [117, 25], has attracted much research attention. It is mainly because, through SSL, the labeling effort can be kept to a minimum, while one can take advantage of a large number of unlabeled samples, which in turn increases the coverage of the sample space and therefore, potential mismatches between training and testing conditions can be reduced.

In this dissertation, we investigate an SSL framework for image concept modeling problems. Image concept modeling is a problem of creating a brief but representative text description for an image by learning associations between visual cues extracted from a collection of images and semantic concepts in human language. Examples are handwritten digit recognition (*image-to-number*), object recognition (*image-to-category*), and image spam detection (*image-to-category*). In general, these problems are very challenging because the amount of randomness in the visual cues is usually very large, thus, requiring (a) a more complex modeling scheme and (b) a large set of labeled images. As for the first issue, we can tackle it by exploiting a nonlinear discriminative learning framework such as a kernelized maximal-figure-of-merit (kMFoM) learning, where we can efficiently build a nonlinear model, optimizing a certain performance metric directly. However, addressing the second issue directly might not be feasible because the time required to label images tends to explode very quickly, as is empirically shown in [76]. As a result, image concept modeling is a great venue where SSL can shine.

Among many possible directions, we focus on semi-supervised *incremental* learning because in practical image concept modeling scenarios, unlabeled images are often collected incrementally. For instance, people usually upload pictures after taking a couple of photos instead of waiting for thousands of images to be gathered. The main concern for semi-supervised incremental learning is how to judicially incorporate unlabeled examples into the learning process while enhancing initially learned concept models over time. In conventional approaches, unlabeled examples are selected when a current model can classify them with

high confidence. The model is then updated using the selected unlabeled samples while their missing class labels are *filled-in* from the prediction results of the current model. This approach, called a confidence score based method, is based on the fact that the probability to generate incorrect classification outputs is minimized if such samples are picked. However, we claim that these samples might not be good candidates to reduce modeling error because they are similar to the image samples already seen in the initial training phase. As a result, even after incorporating a large number of unlabeled samples, we might end up with a sub-optimal concept model, which we refer to as the sub-optimality problem of a confidence score based method. Therefore, in this work, we develop an expected error reduction function that measures the contribution to reducing the classification error of each unlabeled example, and use the expected error reduction function to select unlabeled samples. For reliable estimation of the expected error reduction, we further make use of an ensemble of classifiers that mitigates potential bias and variance of the estimated amount of reduction caused by a small size of initial labeled data.

Another aspect of image concept modeling is that images involve multiple features (e.g., color, texture, shape, etc.). In fact, to be able to handle multiple features is a key to achieving good concept models as studied in [81]. In the context of semi-supervised learning, incorporating multiple features is often referred to as *multi-view learning* [10, 29, 90]. In conventional multi-view learning, individually learned concept models are commonly enforced to be agreed in their predictions on unlabeled samples, known as *agreement assumption*. In this research, we argue that imposing this agreement assumption equally for all unlabeled examples might not always be advantageous to concept modeling as different features might indicate different classes (e.g., color might indicate *cloud,* while texture might indicate *a polar bear.*). Thus, we propose an agreement function that measures how much the agreement assumption should be enforced or relaxed for each unlabeled sample based on the consistency of local information across different views. This agreement function is then embedded into a *reproducing kernel Hilbert space* (RKHS) so that any discriminative kernelized learning frameworks, the one in [18], can be used.

In sum, the research objective of this dissertation is to develop a discriminative semi-supervised incremental learning framework with a multi-view perspective for image concept modeling problems. Various image concept modeling problems including object recognition, handwritten-digit recognition, and image-spam detection are tested to highlight the effectiveness of the proposed framework. The procedures of the proposed framework are summarized as follows: a kernel matrix for an image data set is computed based on some features, such as color, texture, or even pixel value. For a data set with multiple features, we evaluate an agreement function and then embed it into the kernel matrix using the multi-view learning technique described in Chapter 5. Next, classification models are trained on the kernel matrix using a discriminative kernelized learning framework, such as kMFoM learning that we will discuss in Chapter 3. Given the learning models, unlabeled images are selected according to their values of the expected error reduction function as explained in Chapter 4. An existing model is then updated with the newly chosen images until some stopping criteria are met.

Finally, we summarize the contributions of this dissertation as follows:

- A kernelized MFoM learning framework that learns a *nonlinear* class dependent score function while optimizing a performance metric directly is proposed and then tested on several image concept modeling problems (Chapter 3).

- An expected error reduction function is proposed based on a Bayesian decision theory and a novel semi-supervised incremental learning framework that tackles the sub-optimality problem of a confidence score-based method using the expected error reduction function is investigated. Furthermore, the use of an ensemble of classifiers is proposed for robustly estimating expected error reduction. The proposed framework is then applied to a couple of image concept modeling tasks (Chapter 4).

- After studying the validity of the agreement assumption in conventional multi-view learning techniques, an agreement function is proposed so that one can selectively impose the agreement assumption on each unlabeled sample. A closed-form solution for a kernel function that unifies multiple features with the agreement function is also

formulated. The effectiveness of the proposed multi-view learning technique is then verified with an artificially generated data set and an image spam detection problem (Chapter 5).

- An integrated learning framework that combines the semi-supervised incremental learning system discussed in Chapter 4 with the multi-view learning technique presented in Chapter 5 is proposed. The advantages of the integrated framework are then shown in an image spam detection problem (Chapter 6).

1.1 Organization of this dissertation

This dissertation is organized as follows:

In Chapter 2, we provide background knowledge of each of the contributions of this dissertation, including discriminative learning for both supervised and semi-supervised cases, multi-view learning, and incremental learning.

In Chapter 3, we develop a kernelized maximal-figure-of-merit (kMFoM) learning approach and conduct a series of experiments on various image concept modeling problems. The kMFoM learning approach is an example of discriminative learning where a certain performance metric is directly optimized similar to the original MFoM learning approach proposed in [43]. Unlike the original MFoM learning, however, the kMFoM learning approach can take advantage of nonlinear class boundaries. The computational complexity associated with non-linearization of the boundaries is minimized through a subspace distance minimization technique.

In Chapter 4, we investigate a novel semi-supervised incremental learning framework. In particular, to tackle the sub-optimality problem of a conventional confidence score-based technique, an expected error reduction function is proposed with which unlabeled samples are selected based on the amount of potential modeling error reduction. We also show how to update a model parameter. Speficially, we first include the selected unlabeled samples into a training data set and then re-run a learning algorithm given the augmented set of data. We demonstate that a robust estimate of the expected error reduction can be achieved by using an ensemble of classifiers, mainly with kernelized maximal-figure-of-merit (MFoM)

classifiers and spectral clustering-based classifiers. Finally, the effectiveness of the proposed technique is examined with two image data sets, such as the COIL-100 data set and the USPS handwritten digit recognition data set.

In Chapter 5, we develop a multi-view learning technique with an agreement function. First, we validate the conventional use of the agreement assumption (i.e., the classification results from different views should be the same for all unlabeled samples) and claim that such usage might not always be advantageous. Instead, we propose an agreement function that measures the degree of matches we want to impose between individual classifiers. We then provide a kernel function with the agreement function that simplifies several feature spaces into a single reproducing kernel Hilbert space. The effectiveness of the use of an agreement function is demonstrated with artificially generated data sets as well as a real-world data set, namely, the TREC05 spam corpus for image spam detection.

In Chapter 6, a semi-supervised incremental learning algorithm with a multi-view perspective is presented after combining the semi-supervised incremental learning technique discussed in Chapter 4 and the multi-view learning technique provided in Chapter 5. We boost the performance of the initial model by taking advantage of multi-view learning. In semi-supervised incremental learning, having a good initial model is crucial to (a) ensure the convergence property of a learning process, and (b) to improve the performances of a resulting model. With the TREC05 spam corpus, we show the importance of good initial models and present the effectiveness of the combined system.

Finally, we conclude this dissertation in Chapter 7 with a concluding remark and possible future research directions.

Chapter II

BACKGROUND AND RELATED WORK

In this chapter we compile background knowledge relevant to this dissertation by reviewing some of the principles of the related work and discussing the-state-of-the-art techniques. First, we review fundamentals of discriminative learning, covering its mathematical formulation and some existing techniques for discriminative learning in Section 2.1. The main purpose of Section 2.1 is to provide a number of essential concepts related to kMFoM learning, which will be discussed in Chapter 3. In Section 2.2, we lay out existing techniques for discriminative semi-supervised learning (SSL), categorizing them into two major groups according to their underlying assumptions. In Section 2.3, we present relevant incremental learning techniques that build the foundation of semi-supervised incremental learning presented in Chapter 4. In Section 2.4, prior work on multi-view semi-supervised learning is discussed so that we can easily absorb the gist of multi-view learning proposed in Chapter 5.

2.1 Background of discriminative learning

In this section, we review principles of discriminative learning and discuss some existing techniques. In particular, Section 2.1.1 introduces basic mathematical formulation of discriminative learning. Then, Section 2.1.2 presents existing discriminative learning techniques developed from the formulation followed by Section 2.1.3 that discusses some variants of discriminative learning techniques, which aim at a direct optimization of commonly used performance metrics. Finally, in Section 2.1.4, we briefly explain non-linearization of discriminative learning.

2.1.1 Mathematical formulation

The origin of discriminative supervised learning can be traced back to work by Neyman and Pearson in the 1930s about statistical hypothesis testing. In their work, given a ratio

between the likelihood scores of a null hypothesis (e.g., a sample x is in the true class) and that of an alternative hypothesis (e.g., x belongs to one of the competing classes), they showed that an optimal testing strategy was to accept the null hypothesis if the likelihood ratio was greater than a threshold and to reject the hypothesis otherwise [52]. In the late 1930s, Fisher proposed a linear discriminant analysis (LDA) where a linear transformation of a feature space $\mathcal{X}$ that maximized separation between classes was used to classify samples [38].

Typically, discriminative learning is formulated as an optimization problem with a discriminant function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and a decision rule $\delta : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is a feature space and $\mathcal{Y} = \{1, \ldots, C\}$ is a space of class labels. More precisely, suppose a discriminant function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and a decision rule $\delta : \mathcal{X} \to \mathcal{Y}$ are defined by

$$f(x, y; \boldsymbol{\theta}) = g_y(x) - g_{y^-}(x), \tag{1}$$

and

$$\delta(x; \boldsymbol{\theta}) = \{y | f(x, y; \boldsymbol{\theta}) > 0, y \in \mathcal{Y}\}, \tag{2}$$

respectively, where $g_y(x)$ is a shorthand notation of a class dependent score function $g(x; \theta_y)$ for a sample $x \in \mathcal{X}$ and its associated class $y \in \mathcal{Y}$, and $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_C]$ denotes a collection of parameters of $g_y(x)$ for $y \in \mathcal{Y}$.

Then, for a labeled set, $\mathcal{L} = \{l_i = (x_{l_i}, y_{l_i}) | x_{l_i} \in \mathcal{X}, y_{l_i} \in \mathcal{Y}, i = 1, \ldots, N_l\}$, a discriminative supervised learning problem is given by

$$\min_{\boldsymbol{\theta} \in \Theta} \frac{1}{N_l} \sum_{i=1}^{N_l} V(\delta(x_{l_i}; \boldsymbol{\theta}), y_{l_i}) + \lambda R(f(\cdot; \boldsymbol{\theta})), \tag{3}$$

where $V : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$ is a loss function that measures the amount of error incurred by the decision rule δ, and $R(f(\cdot; \boldsymbol{\theta}))$ is a regularization term that prevents f from being too complex. Moreover, λ is a positive constant that controls the significance of the regularization term. Because the first term of Eq. (3) is usually called an *empirical error* on a training data set $\mathcal{D}$, the minimization in Eq. (3) can be interpreted as finding a model that fits well to the training data while keeping the complexity of the model as small as possible.

In Eq. (1), the definition of y^- can vary depending on the type of a problem. In particular, in multi-class (i.e., $C > 2$) and binary-class (i.e., $C = 2$) classification problems,

$y^- = \arg\max_{y' \in \mathcal{Y} \setminus y} g_{y'}(x)$. On the other hand, in multi-label problems (i.e., x can belong to multiple classes at the same time.), $y^- = \mathcal{Y} \setminus y$. Note that regardless of the choice of y^- for Eq. (1), the discriminant function f measures separation between $g_y : \mathcal{X} \to \mathbb{R}$, a class dependent score function for a class y, and g_{y^-}, a class dependent score function for a competing class of y. Therefore, it can be said that any learning methods that maximize separation between one class from the other can be thought of as discriminative learning techniques as seen in Fisher's work.

2.1.2 Existing discriminative learning techniques

Several discriminative learning algorithms have been proposed based on the optimization problem in Eq. (3). Examples include support-vector machines (SVMs) [99], logistic regression [48, 11], Gaussian processes (GPs) [77], etc. The main difference among them is the definition of the loss function V. Specifically, SVMs use a hinge-loss function given by

$$\max(0, 1 - f(x_{l_i}, y_{l_i}; \boldsymbol{\theta})), \tag{4}$$

while logistic regression uses a linear logit function embedded into a sigmoid function given by

$$\log(1 + e^{-f(x_{l_i}, y_{l_i}; \boldsymbol{\theta})}), \tag{5}$$

and GPs exploits an L_2-loss function given by

$$(y_{l_i} - f(x_{l_i}, y_{l_i}; \boldsymbol{\theta}))^2. \tag{6}$$

On the other hand, various $R(f(\cdot; \boldsymbol{\theta}))$s have also been creating many discriminative learning approaches. For example, Lasso [96] and L_1-normed SVMs [116, 102] define an L_1-norm of $\boldsymbol{\theta}$ for $R(f(\cdot; \boldsymbol{\theta}))$ to achieve a sparse parameter vector $\boldsymbol{\theta}$. In contrast, regular SVMs [99] and ridge regression [48] use a squared L_2-norm of $\boldsymbol{\theta}$ for $R(f(\cdot; \boldsymbol{\theta}))$ for a mathematical convenience due to the convexity of a L_2-norm. Other possibilities are an $L_{1,\infty}$-norm given by $\|\boldsymbol{\theta}\|_{1,\infty}$ [75], and an L_p-norm of the second derivative of $f(\cdot; \boldsymbol{\theta})$ [48].

When a class label y is a sequence of classes with some structural dependencies, we typically rely on a class-posterior probability $P(y|x, \boldsymbol{\theta})$ instead of the discriminative function

$f(\cdot;\boldsymbol{\theta})$ in Eq. (1). This is because one can take advantage of a conditional independence among y when defining $P(y|x;\boldsymbol{\theta}^t)$ so that an associated optimization problem can be simplified significantly. More precisely, suppose there is a conditional independence among y given x and $\boldsymbol{\theta}$. Then, a discriminative supervised learning problem can be reformulated using $P(y|x,\boldsymbol{\theta})$ and $P(\boldsymbol{\theta};\nu)$, a prior density $P(\boldsymbol{\theta};\nu)$ of $\boldsymbol{\theta}$, as follows:

$$\max_{\boldsymbol{\theta}\in\Theta} P(\boldsymbol{\theta}|\mathcal{L};\alpha) \quad \Leftrightarrow \quad \max_{\boldsymbol{\theta}\in\Theta} \frac{\prod_{i=1}^{n_l}\prod_{k=1}^{m} P_k(y_{l_i}|x_{l_i},\boldsymbol{\theta})P(\boldsymbol{\theta};\nu)}{\int \prod_{i=1}^{n_l}\prod_{k=1}^{m} P_k(y_{l_i}|x_{l_i},\boldsymbol{\theta})P(\boldsymbol{\theta};\nu)d\boldsymbol{\theta}} \tag{7}$$

$$\Leftrightarrow \quad \max_{\boldsymbol{\theta}\in\Theta} \prod_{i=1}^{n_l}\prod_{k=1}^{m} P_k(y_{l_i}|x_{l_i},\boldsymbol{\theta})P(\boldsymbol{\theta};\nu), \tag{8}$$

where $P(y|x,\boldsymbol{\theta})$ is factored into m components as $P(y|x,\boldsymbol{\theta}) = \prod_{k=1}^{m} P_k(y|x,\boldsymbol{\theta})$ as a result of the conditional independence among y, and ν is a hyper-parameter of $P(\boldsymbol{\theta};\nu)$. Examples of such discriminative learning approaches for structural class label cases are Markov random fields (MRFs) [59], conditional random fields (CRFs) [61], hidden-conditional random fields (H-CRFs) [104], and many more.

Note that one can derive an equivalence between the optimization problem given in Eq. (8) and that in Eq. (3). To see this equivalence, suppose $P(y_{l_i}|x_{l_i},\boldsymbol{\theta})$ and $P(\boldsymbol{\theta};\nu)$ are defined as follows:

$$P(y_{l_i}|x_{l_i},\boldsymbol{\theta}) = \frac{1}{Z(x_{l_i},\boldsymbol{\theta})} e^{-V(f(x_{l_i},y_{l_i};\boldsymbol{\theta}),y_{l_i})} \tag{9}$$

$$P(\boldsymbol{\theta};\nu) = \frac{1}{Z(\nu)} e^{-\lambda R(f(\cdot;\boldsymbol{\theta}))}, \tag{10}$$

where $Z(x_{l_i},\boldsymbol{\theta})$ and $Z(\nu)$ are normalization components. Then, if we take the logarithms of $P(y_{l_i}|x_{l_i},\boldsymbol{\theta})$ and $P(\boldsymbol{\theta};\nu)$ in Eq. (10) and in Eq. (9), and then plug them into Eq. (8), respectively, it can be seen that Eq. (8) becomes the same as Eq. (3).

2.1.3 Performance metrics and discriminative learning

So far, we have discussed commonly used discriminative supervised learning techniques based on the principle of maximizing separation between classes. There are some variants of discriminative supervised learning techniques that focus on optimizing performance metrics directly. Some of the commonly used performance metrics are listed in Table 1.

The underlying idea of these techniques is that by matching the loss function V used in training to the metric used for actual evaluation, one can reduce mismatches between

Table 1: The definitions of some commonly used performance metrics. For simplicity, we assume that there are only two classes; positive vs. negative. Note that precision and recalls are computed with respect to the positive class. *no.* stands for *number*

Classification error	$\frac{\text{the total no. of samples} - \text{the no. of correctly classified samples}}{\text{the total no. of samples}}$
Precision	$\frac{\text{the no. of correctly classified positive samples}}{\text{the total no. of positive samples}}$
Recall	$\frac{\text{the no. of correctly classified positive samples}}{\text{the no. of samples classified as positive}}$
F_1-measure	$\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$

training conditions and testing conditions. One of the earlier examples of these techniques was a minimum classification error (MCE) learning algorithm used in speech recognition [57]. In MCE learning, the classification error is approximated using a class $0-1$ loss function l given by

$$l(x_{l_i}, y_{l_i}; \boldsymbol{\theta}) = \frac{1}{1 + e^{\alpha f(x_{l_i}, y_{l_i}; \boldsymbol{\theta}) + \beta}}, \tag{11}$$

where α and β are parameters to tune. With Eq. (11), it can be seen that as the value of a discriminant function for x_{l_i}, $f(x_{l_i}, y_{l_i}; \boldsymbol{\theta})$ becomes larger and larger, the value of the class $0-1$ loss function gets smaller and smaller, which conforms to the decision rule defined in Eq. (2). Therefore, the classification error, denoted as $\text{Err}(\boldsymbol{\theta})$, can be approximated with a continuous and differentiable function l as

$$\text{Err}(\boldsymbol{\theta}) = \sum_{i=1}^{N_l} l(x_{l_i}, y_{l_i}; \boldsymbol{\theta}), \tag{12}$$

which thus can be optimized with a standard non-constrained optimization tool, such as a generalized probabilistic decent (GPD) algorithm. More recently, variants of SVMs were proposed to maximize performance metrics, such as average precision (AP), F_1-measure, etc., as in [16, 55]. One possible drawback of these SVM-based techniques, however, is that it is a lower bound of a performance metric that is actually maximized. This lower bound might not be tight enough especially when a chosen performance metric and a regularization term are not properly normalized. As an alternative method, a maximal-figure-of-merit (MFoM) learning approach was proposed in which direct optimization of a commonly used performance metric was considered [43]. Similar to MCE learning, an MFoM learning algorithm often makes use of the class $0-1$ loss function l given in Eq. (11). The main difference

between an MFoM learning technique and the MCE learning algorithm is the fact that in MFoM learning, a designer has an option to choose performance metric of interest depending on an application. In contrast, in MCE, only the classification error can be optimized. Nevertheless, the effectiveness of the existing MFoM learning approach is somewhat limited by the fact that a class dependent score function g can only be linear.

2.1.4 Non-linearization of discriminative learning techniques

A nonlinear class dependent score function can be obtained by applying a feature map, $\Psi : \mathcal{X} \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space. A simple example of such a feature map is when x is a two-dimensional vector, such as $x = [x_1, x_2]^T$. Then, we can define Ψ as

$$\Psi : (x_1, x_2) \to (x_1^2, \sqrt{2}x_1x_2, x_2^2), \tag{13}$$

which results in an oval-shaped class dependent score function. Typically, the size of a collection of possible nonlinear score functions is reduced by using a *reproducing kernel Hilbert space* (RKHS), $\mathcal{H}_K$. According to the Moore-Aronszajn theorem [2], for every $\mathcal{H}_K$, there exists an associated kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ given by $k = \langle \Psi(x), \Psi(y) \rangle_{\mathcal{H}_K}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}_K}$ represents an inner product in $\mathcal{H}_K$ and $\Psi(x)$ is a function in $\mathcal{H}_K$ associated with x. Similarly, for every kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there also exists an associated reproducing kernel Hilbert space whose inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_K}$ is equal to the value of the kernel function. Usually, the inner product $\langle f, g \rangle_{\mathcal{H}_K}$ for f,$g \in \mathcal{H}_K$ is defined as

$$\int f(t)g(t)dt, \tag{14}$$

when f and g are assumed to be a function of t. There have been a number of studies focusing on non-linearization of class dependent score functions, such as [84, 33, 77, 99].

2.2 Background of discriminative semi-supervised learning (SSL)

In this section we describe the background knowledge for discriminative semi-supervised learning (SSL). In essence, discriminative SSL shares the same spirit of regular discriminative supervised learning; to find a decision rule in such a way that separation between classes is maximized. In discriminative SSL however, one also needs to identify an effective way to

take advantage of unlabeled samples when learning a classification model. In this section, we give examples of how unlabeled samples can be used in discriminative SSL, starting from a brief introduction of its mathematical formulation in Section 2.2.1. We then go over some of the state-of-the-art techniques proposed in the literature in Sections 2.2.2 and 2.2.3, categorizing them by their treatment of unlabeled samples.

2.2.1 Mathematical formulation

Let $\mathcal{D} = \mathcal{L} \cup \mathcal{U}$ be a training data set with N data samples ($N = N_l + N_u$), where $\mathcal{L}$ is an labeled data set given by $\mathcal{L} = \{l_i = (x_{l_i}) | x_{l_i} \in \mathcal{X}, i = 1, \ldots, N_l\}$ and $\mathcal{U}$ is an unlabeled data set given by $\mathcal{U} = \{u_i = (x_{u_i}) | x_{u_i} \in \mathcal{X}, i = 1, \ldots, N_u\}$. Then, discriminative SSL algorithms can typically be written as an optimization problem given by

$$\max_{\boldsymbol{\theta} \in \Theta} P(\boldsymbol{\theta} | \mathcal{D}, \mu; \nu) \Leftrightarrow \max_{\boldsymbol{\theta} \in \Theta} \prod_{i=1}^{N_l} P(y_{l_i} | x_{l_i}, \boldsymbol{\theta}) P(\boldsymbol{\theta} | \mu; \nu) \prod_{i=1}^{N} P(x_i | \mu), \tag{15}$$

where $P(x|\mu)$ is a marginal distribution of x parametrized with μ, $P(\boldsymbol{\theta}|\mu;\nu)$ is an additional probability distribution for $\boldsymbol{\theta}$ given μ and ν, and the subscripts i and l_i are used to denote i^{th} sample in $\mathcal{D}$ and i^{th} sample in $\mathcal{L}$, respectively. In other words, in discriminative SSL, we are looking for a maximizer of the likelihood score computed from training samples, including both labeled and unlabeled sets. Although a high-level concept of Eq. (15) is similar to that of Eq. (8), a key innovation of Eq. (15) is the introduction of the additional probability distribution $P(\boldsymbol{\theta}|\mu;\nu)$. With this distribution function, $\boldsymbol{\theta}$ now becomes a function of both ν and μ whereas in Eq. (8), $\boldsymbol{\theta}$ is a function of only ν. In fact, this dependency is what makes unlabeled samples valuable in discriminative SSL. Without $P(\boldsymbol{\theta}|\mu;\nu)$, it can be easily seen that unlabeled samples have no effect on estimating $\boldsymbol{\theta}$, which is also pointed out in [85]. In many cases however, $P(\boldsymbol{\theta}|\mu;\nu)$ is not explicitly modeled due to mathematical intractability. Instead, most discriminative SSL techniques take advantage of a simple dependency assumption between μ and $\boldsymbol{\theta}$, which will be discussed in the following two sections.

2.2.2 Methods using low-density separation

In semi-supervised learning it is generally assumed that samples in a single cluster share the same class label. We have a toy example that satisfies such an assumption in Figure 1, where the marginal distribution of x, $P(x|\mu)$ is represented as two Gaussian mixtures, each of which has two mixture components. In Figure 1, a class boundary that passes in-between

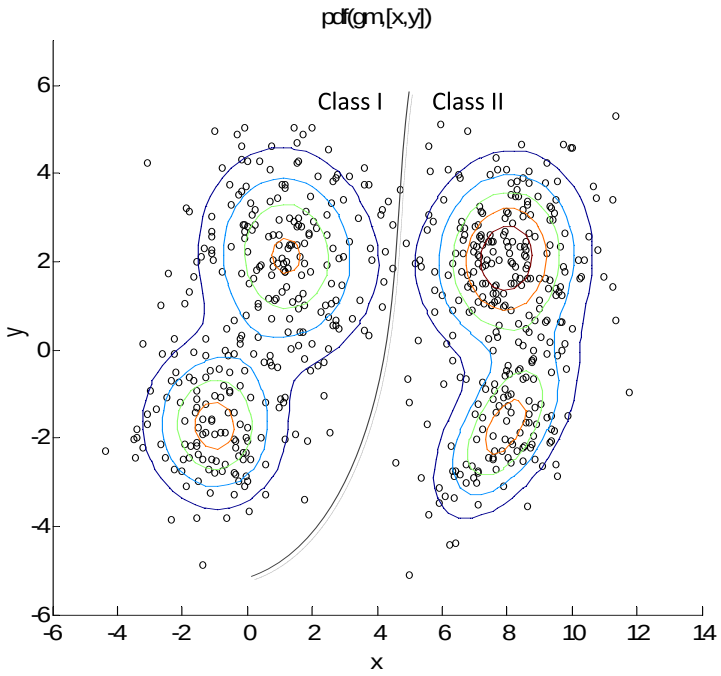


Figure 1: An illustration of a low-density separation assumption. Two Gaussian mixtures are separated by a decision boundary that passes through in-between of two mixtures. The values of a marginal distribution of x, $P(x|\mu)$, are represented using iso-contours; red corresponds to the highest value and black corresponds to the lowest value. According to the low-density separation assumption, class boundaries should be placed where the marginal distribution of x is low. Here, it is seen that the low-density separation assumption holds.

two Gaussian mixtures can be found, which makes each cluster in Figure 1 labeled as the same class. Examining the iso-contours of the marginal distribution, it can be also found that the boundary is located where the probability of x is small, which is the reason why such an assumption is called *a low-density separation assumption* in the literature.

One of the earlier techniques that exploited this low-density separation assumption successfully was transductive SVMs (TSVMs) and its variants [56, 24]. In TSVMs, an extra

loss function given by

$$\sum_{i=1}^{n_u} \max(1 - f(x_{u_i}, \delta(x_{u_i}; \boldsymbol{\theta})), 0) \tag{16}$$

was minimized along with a hinge-loss function defined in Eq. (4). Because Eq. (16) prefers having class boundaries such that all unlabeled samples are located outside of a margin, the smallest distance of training samples from the boundaries, one can see that the low-density separation assumption will be automatically satisfied.

A more recent technique proposed in [46] made use of a conditional entropy of y given x to exploit the assumption. To be more specific, in [46], an optimization problem was solved in a way that the conditional entropy of y given x computed over $\mathcal{U}$ was minimized, while the likelihood score of $P(y|x, \boldsymbol{\theta})$ over $\mathcal{L}$ was maximized. Conceptually, the smaller conditional entropy is, the less uncertain $P(y|x, \boldsymbol{\theta})$ becomes. Thus, by minimizing the conditional entropy, class boundaries will be placed where unlabeled samples are sparse, thus satisfying the low-density separation assumption. Another technique based on this assumption is also presented in [27]. In [27], the low-density separation assumption was imposed by stipulating local behaviors of $P(y|x, \boldsymbol{\theta})$ for unlabeled samples. In particular, let $\mathcal{N}_{x_{u_i}}$ be a set of neighboring samples of x_{u_i}, the i^{th} unlabeled sample in $\mathcal{U}$. Then the value of $P(y_{u_i}|x_{u_i}, \boldsymbol{\theta})$ has to be as close as possible to the value of $P(y|x, \boldsymbol{\theta})$ for all x in $\mathcal{N}_{x_{u_i}}$, which in turn, the value of $P(y|x, \boldsymbol{\theta})$ can only be changed when $P(x|\mu)$ is low.

2.2.3 Methods using a data-driven kernel

In some semi-supervised learning techniques, unlabeled samples are used to create a kernel matrix that determines closeness among data points. In this section, we review techniques that make use of such *data-driven kernel*s. The first set of examples is techniques exploiting some manifold structures in training data. In some applications, it is considered that the support of the marginal distribution $P(x|\mu)$ is in a low-dimensional manifold. Figure 2 shows a simple case of this, where a set of data samples is scattered randomly in a seemingly complex pattern in their original space while the actual support for the samples is a perfect circle in a two-dimensional space. In this case one can take advantage of a *manifold assumption*. The manifold assumption states that samples close to each other along

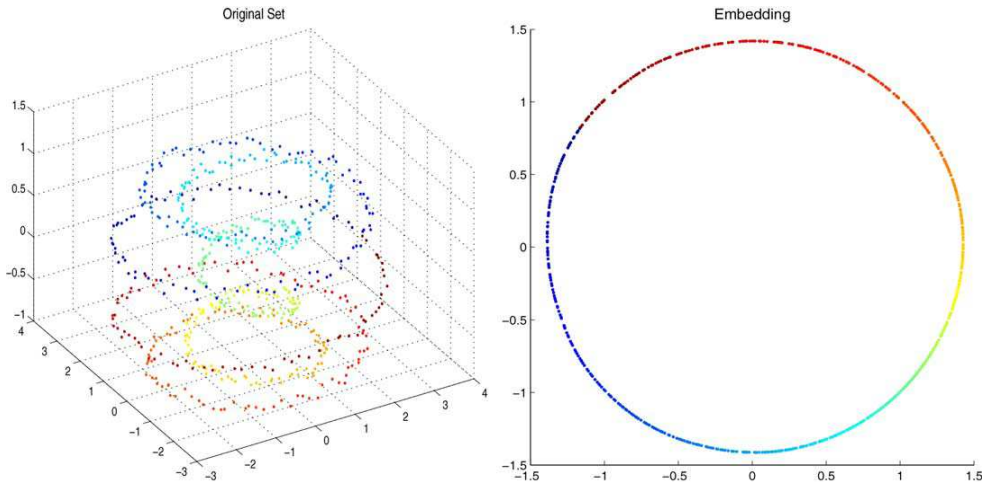


Figure 2: An illustration of a manifold embedded in a higher-order space, w.hich is obtained from [26]. Note that on the left, data are scattered in a rather complex pattern in a three-dimensional space. In contrast, data samples form a perfect circle along the manifold in a two-dimensional space on the right.

the manifold should have similar class labels. Because this assumption is closely related to manifold learning, an unsupervised learning technique that is often solved by using a graph Laplacian L [47], in [12, 7, 54, 114], a graph Laplacian L has also been used extensively for solving discriminative SSL problems. To see this, suppose there is a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a vertex set corresponding to a training data set $\mathcal{D}$, and $\mathcal{E}$ is an edge set encoding pairwise relations between vertices with associated weights w. Now, given v_i and v_j in $\mathcal{V}$, the i^{th} and the j^{th} samples in $\mathcal{D}$, respectively, let $e_{ij} \in \mathcal{E}$ be an edge between them and w_{ij} be an weight associated with e_{ij}, which represents closeness between v_i and v_j. Then, the manifold assumption is imposed for discriminative SSL by defining an optimization problem given by

$$\min_{f} \sum_{y \in \mathcal{Y}} \sum_{i,j=1}^{N} w_{ij} (f(x_i, y; \boldsymbol{\theta}) - f(x_j, y; \boldsymbol{\theta}))^2, \tag{17}$$

where N is the total number of training samples including labeled and unlabeled data, $f(x, y; \boldsymbol{\theta})$ is a discriminant function defined in Eq. (1), and $\mathcal{Y}$ is a set of class labels. Therefore, given the minimizer of Eq. (17), the values of the discriminant function for the i^{th} and j^{th} samples, denoted as $f^*(x_i, y; \boldsymbol{\theta})$ and $f^*(x_j, y; \boldsymbol{\theta})$, respectively, should be similar if those samples are close to each other (i.e., w_{ij} is large.) along the graph G. Recently, it has been pointed out that Eq. (17) could be considered as a squared *semi*-L_2-norm of f

with respect to a graph Laplacian L [91]. Based on this observation, a data-driven kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ can be constructed in a way that the value of k increases as two data points get closer and closer in a manifold as discussed in [91].

Another technique that makes use of the manifold assumption is a label propagation technique presented in [25]. In essence, the label propagation technique shares the same formulation as in Eq. (17) because similarly defined weights w_{ij} control the maximum amount of label information that can be propagated between two samples x_i and x_j. Thus, a similar data-driven kernel k will be created in the end. One advantage to using label propagation techniques over the techniques presented in [12, 7, 54, 114] is that in label propagation techniques, not only undirected graphs, but also directed graphs can be used, ensuring that the label information is propagated only from labeled samples to unlabeled samples, not the other way around [25].

Other examples to learn a data-driven kernel function are based on a geometry of $P(x|\mu)$. In particular, in [86], unlabeled samples were assumed to be drawn from a mixture of probability distributions. Then, a kernel function k was defined using mutual information such that $k(x_i, x_j) \geq k(x'_i, x'_j)$ for all $x_i, x_j, x'_i, x'_j \in \mathcal{X}$, when x_i and x_j are in the same mixture component, while x'_i and x'_j are in different mixture components. In other words, a kernel function is created in a way that the value is large when two samples are deemed to be in the same mixture component, and it is small if the sample are placed in different components. A similar technique was proposed in [70]. Specifically, given a Gaussian mixture with m mixture components, a kernel function $k(x_i, x_j)$ for x_i and x_j, called *Fisher kernel*, is defined as

$$k(x_i, x_j) = \sum_{k=1}^{m} P(\pi_k|x_i)P(\pi_k|x_j)x_i^T\Sigma_k^{-1}x_j, \tag{18}$$

where $P(\pi_k|x_i)$ and $P(\pi_k|x_j)$ are the probabilities of x_i and x_j belonging to the k^{th} mixture component π_k, respectively, and Σ_k is a covariance matrix of the k^{th} mixture component. By inspection, one can easily see that the *Fisher kernel* also satisfies the inequality; $k(x_i, x_j) \geq k(x'_i, x'_j)$ for all $x_i, x_j, x'_i, x'_j \in \mathcal{X}$, when x_i and x_j are in the same mixture, while x'_i and x'_j are in different mixtures.

2.3 Relevant incremental learning techniques

All techniques describedin Sections 2.2 and 2.2 can be categorized as batch learning because when training a model, a large number of training data are used as a whole. In real-world applications however, such large sets of training samples are rarely prepared before learning a classifier. In contrast, in many cases, a handful of examples are constantly collected over time while a classification system is operating. A simple example is a face recognition system that can create tags for people in a photo repository automatically. When building such a system, people tend to update the repository with tens of pictures daily, not with thousands of photos after waiting for months. Therefore, it is desired to perform training incrementally, enhancing an existing system by incorporating the small training data samples, which is the main topic of this section, known as *incremental learning*. Figure 3 illustrates a system overview of a typical incremental learning framework. In Figure 3, it can be seen that a

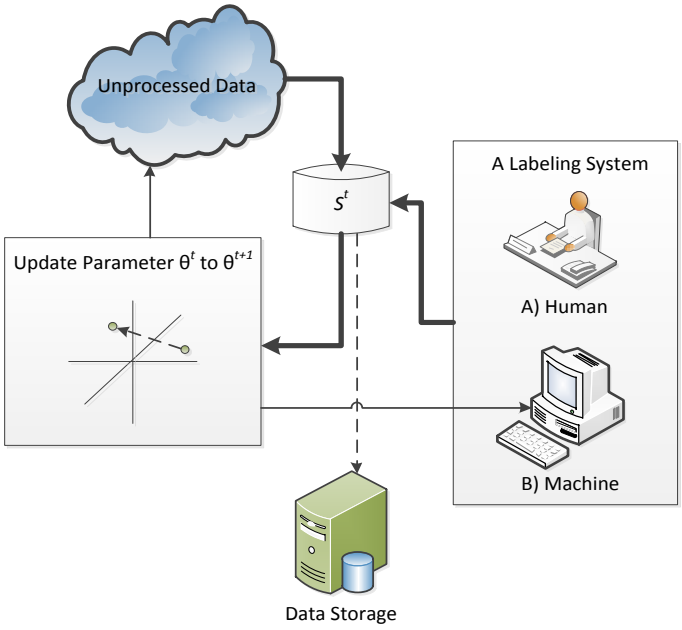


Figure 3: An overview of incremental learning. Given a collection of unprocessed data, a selection set $\mathcal{S}^t$ is created. After class labels for $\mathcal{S}^t$ are generated either by humans or machines, a parameter vector $\boldsymbol{\theta}^t$ is updated to $\boldsymbol{\theta}^t$. When the labeling system consists of machines, this updated parameter is then used to enhance the labeling mechanism. Note that sometimes, the labeling system helps to create $\mathcal{S}^t$ to speed up the convergence of $\boldsymbol{\theta}$.

parameter vector at time t, $\boldsymbol{\theta}^t$, is updated to $\boldsymbol{\theta}^{t+1}$ with a selection set $\mathcal{S}^t$, a collection of data samples at time t chosen from unprocessed data. In the meantime, it is also seen that class labels for the samples in $\mathcal{S}^t$ are generated by a certain labeling system that can be either humans or machines. Additionally, the parameter $\boldsymbol{\theta}^t$ helps to create $\mathcal{S}^t$ in such a way that the update of the parameter vector $\boldsymbol{\theta}^t$ to $\boldsymbol{\theta}^{t+1}$ is not detrimental.

From the analysis of Figure 3, it can be seen that there are two main research issues regarding development of an incremental learning framework: (a) how to update the parameter vector efficiently, and (b) how to construct $\mathcal{S}^t$ ensuring classification models are enhanced at each iteration. In the literature, the first issue is studied mostly in fully supervised cases as in [31, 100], while the second issue has mainly been explored in semi-supervised settings. As the theme of this dissertation is to develop a semi-supervised learning framework in this section, we will primarily describe techniques to build $\mathcal{S}^t$ from a unlabeled data set. In particular, in Section 2.3.1, we discuss active learning techniques, where a user (or an oracle) provides the ground-truth label information (i.e., a man-powered labeling system) for unlabeled samples in $\mathcal{S}^t$. In Section 2.3.2, we presents semi-supervised incremental learning techniques, where labels for unlabeled samples in $\mathcal{S}^t$ are predicted automatically by machines (i.e., a machine-driven labeling system). Nevertheless, active learning and semi-supervised incremental learning share similar mechanical procedures, as shown in Figure 3 and thus, comparisons between them can provide useful insights to improve either technique. We will see such a benefit in Chapter 4 when presenting our semi-supervised incremental learning algorithm.

2.3.1 Active learning

In active learning, a classification system invokes queries to users to fill in the missing class labels for $\mathcal{S}^t$ [88]. Because the cost of such inquiries is usually very high, the main issue of active learning is to minimize the number of those queries while retaining the modeling accuracy as close as possible to the case when all possible training data are used. One of the earlier active learning techniques is based on *uncertainty-based sampling* proposed by David D. Lewis in the '90s [64]. There, the system asked class information to users for unlabeled

samples that a current model was the least confident about their class labels. Because the level of uncertainty was evaluated based on a class posterior probability $P(y|x;\boldsymbol{\theta}^t)$, $\mathcal{S}^t$ comprised unlabeled samples such that $\max_{y\in\mathcal{Y}} P(y|x;\boldsymbol{\theta}^t)$ is small. Recently, information-theoretic justifications of this uncertainty-based sampling strategy were provided in [50, 89]. In particular, it was shown that an unlabeled example with the maximum entropy would provide the maximum information gain if the class label for that sample is revealed. So, an unlabeled sample x from which the maximum entropy is achieved should be selected, which is the sample that $\max_{y\in\mathcal{Y}} P(y|x;\boldsymbol{\theta}^t)$ is the smallest as claimed in [64]. On the other hand, SVM-based active learning techniques can also be considered as variants of the uncertainty-based sampling scheme [98, 97, 65]. A typical strategy of SVM-based active learning is as follows: to select a sample with the smallest margin, the shortest distance among training samples from decision boundaries. This use of a margin value is based on the fact that such a sample will reduce an *empirical error* maximally as claimed in [65]. Later, in [49], an extension of SVM-based active learning was also studied, where an efficient batch selection algorithm was proposed using quadratic programming (QP) and sub-modular functions.

Query-by-committee (QBC) [41, 1, 67] is another popular technique for active learning. In the QBC, the members in a committee usually consisted of several discriminative classifiers who should select a sample x such that their prediction results disagree maximally (e.g., a half of the committee members classify x as positive, but the remaining half determines x to be negative.). To understand the underlying theoretical foundation of the QBC, there are a couple of concepts that need to be introduced: (a) a version space, and (b) consistency to a data set. A version space is a region in a feature space $\mathcal{X}$ where committee members do not share their decisions. On the other hand, committee members are said to be *consistent with* a data set if every system in the committee attains a perfect modeling accuracy to the data. Then, given a consistent committee to labeled samples, it was proved, [41, 1, 67], that minimizing the size of the version space would reduce modeling error maximally and thus, justifying the QBC criterion.

In [80, 118], algorithms based on an *expected error reduction* were proposed to reduce classification error directly. The essence of these techniques is to choose a sample x such

that a classification error at time $t+1$ is maximally reduced compared to that at time t. Because the ground-truth label for x is unknown at the time of selection (it will be provided by users after the selection), the amount of an error reduction is averaged over a posterior probability of the label y given x, $P(y|x)$, instead. Therefore, given an unlabeled data set available at time t, $\mathcal{U}^t$, a sample $x_{u_i^*}$ is selected as follows:

$$x_{u_i^*} = \underset{x_{u_i} \in \mathcal{U}^t}{\arg\min} \sum_{y \in \mathcal{Y}} P(y|x_{u_i})[V(\delta(x_{u_i}; \boldsymbol{\theta}^t(x_{u_i})), y)], \tag{19}$$

where $\boldsymbol{\theta}^t(x_{u_i})$ denotes a parameter vector at time t learned from $\mathcal{L}^t$, a labeled data set available at time t, in addition to an unlabeled sample in $\mathcal{U}^t$, x_{u_i}, (i.e., $\mathcal{L}^t \cup \{x_{u_i}\}$), and V is a loss function as usual. Note that the parameter vector $\boldsymbol{\theta}^t(x_{u_i})$ is a function of a sample in $\mathcal{U}^t$ because depending on the sample chosen, different parameters will be learned. This implies that solving Eq. (19) is often computationally prohibitive because one might have to estimate parameters for all samples in $\mathcal{U}^t$ to solve Eq. (19). As a result, expected error reduction based techniques have been applied to only a few cases where efficient algorithms to estimate $\boldsymbol{\theta}^t(x_{u_i})$ exist.

2.3.2 Semi-supervised incremental learning

As mentioned earlier, in semi-supervised incremental learning, the ground-truth label information for $\mathcal{S}^t$ has to be estimated somehow by the learning algorithm itself. One naive approach to addressing this issue is to simply trust the outputs that a current classification model supplies. More formally, suppose $\delta(x; \boldsymbol{\theta}^t)$ is a decision rule for x learned with $\mathcal{L}^t$, a labeled set at time t. The prediction results given by $\delta(x; \boldsymbol{\theta}^t)$ for $\forall x \in \mathcal{S}^t$ are then treated as clean, ground-truth class labels. In many cases, however, the number of samples used to train the decision rule is often very small and thus, the generated label information is rather noisy. Moreover, when such noisy class labels are aggregated, it is likely that the classification system will diverge (i.e., the final model performs worse than the initial model) eventually.

To avoid this failure, in the '90s, a confidence score function - a measure of confidence that an existing model had regarding its decision $\delta(x; \boldsymbol{\theta}^t)$ on a sample x - was explored in [110, 87]. According to a theory of hypothesis testing, samples with high confidence

scores would have a low probability of making incorrect predictions. Thus, the strategy was to incorporate unlabeled samples with the high confidence scores and their predicted class labels into $\mathcal{S}^t$, namely a confidence score based method. Algorithm 1 presents this confidence score based method in detail. One problem of this technique is however, that it

Algorithm 1 Confidence score-based semi-supervised incremental learning

prepare $\mathcal{U}^0$ and $\mathcal{L}^0$
initialize $\boldsymbol{\theta}^0$ with $\mathcal{L}^0$
$t \leftarrow 0$
repeat
  compute confidence scores $s_c(\delta(x; \boldsymbol{\theta}^t))$ for $x \in \mathcal{U}^t$
  $N_u^t \leftarrow |\mathcal{U}^t|$
  $k_u^t \leftarrow$ the number of samples to be selected at time t
  $\tilde{\mathcal{U}}^t \leftarrow \{x_{(i)} \mid x_{(i)} \in \mathcal{U}^t,\ s_c(\delta(x_{(1)}; \boldsymbol{\theta}^t)) \geq s_c(\delta(x_{(2)}; \boldsymbol{\theta}^t)) \geq \cdots \geq s_c(\delta(x_{(N_u^t)}; \boldsymbol{\theta}^t))\}$
  $\mathcal{S}^t \leftarrow \{(x_{(i)}, y_{(i)}) \mid i \leq k_u^t,\ y_{(i)} = \delta(x_{(i)}; \boldsymbol{\theta}^t),\ x_{(i)} \in \tilde{\mathcal{U}}^t\}$
  $\mathcal{L}^{t+1} \leftarrow \mathcal{L}^t \cup \mathcal{S}^t$
  $\mathcal{U}^{t+1} \leftarrow \mathcal{U}^t \backslash \mathcal{S}^t$
  update $\boldsymbol{\theta}^{t+1}$ with $\mathcal{L}^{t+1}$
  if $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t$ **then**
    break
  else
    $t \leftarrow t + 1$
  end if
until $|\mathcal{U}^t| = 0$

might result in a suboptimal solution even after incorporating a large number of unlabeled samples chosen in order of their confidence scores (i.e., $\boldsymbol{\theta}^t \approx \boldsymbol{\theta}^0$ for all t, where $\boldsymbol{\theta}^0$ represents an initial parameter vector). This is mainly due to the fact that highly confident samples are often similar to already processed samples and thus, have insignificant effects on changing decision boundaries. Recent work, such as [93, 34, 17], also revealed the sub-optimality problem of a confidence score-based technique empirically. Moreover, as identified in [113], selecting samples in order of confidence scores does not guarantee choosing samples according to their marginal distribution $P(x|\mu)$. This implies that the possibility for a confidence score based technique to suffer from a *class imbalance problem* could be large.

On the other hand, the risk of working with incorrect class labels during an incremental learning process can also be reduced using an expectation-maximization (EM) algorithm. Because with an EM algorithm, a selection set $\mathcal{S}^t$ is no longer constructed, a class posterior

probability $P(y|x;\boldsymbol{\theta}^t)$ computed for every sample is now used for parameter updates. In particular, as in [69, 35], a new parameter vector at time $t+1$, $\boldsymbol{\theta}^{t+1}$, will be obtained such that $\boldsymbol{\theta}^{t+1}$ is a maximizer of an *auxiliary function* $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t)$ as

$$\begin{aligned}\boldsymbol{\theta}^{t+1} &= \underset{\boldsymbol{\theta}=[\theta_1,\ldots,\theta_C]\in\Theta}{\arg\max}\ Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t) \qquad (20)\\ &= \underset{\boldsymbol{\theta}=[\theta_1,\ldots,\theta_C]\in\Theta}{\arg\max}\ [\sum_{i=1}^{N_l}\log P(y_{l_i})P(x_{l_i}|y_{l_i};\theta_{y_{l_i}}) \\ &\qquad + \sum_{i=1}^{N_u}\sum_{y\in\mathcal{Y}} P(y|x_{u_i};\boldsymbol{\theta}^t)\log P(y)P(x_{u_i}|y;\theta_y)], \qquad (21)\end{aligned}$$

where $P(x_{l_i}|y_{l_i};\theta_{y_i})$ (or $P(x_{u_i}|y;\theta_y)$) denotes a class conditional probability of the i^{th} labeled (or unlabeled) sample given a class label y_{l_i} (or y), and $P(y_{l_i})$ (or $P(y)$) represents a class prior distribution for y_{l_i} (or y). In earlier work, a very simple model for the class conditional probability (e.g., a single Gaussian distribution in [69]) has been used because of computational intractability. More recently, in [108, 109], by applying linear discriminant analysis (LDA) at every iteration in the EM procedures, complex models were exploited by reducing the dimension of a feature space $\mathcal{X}$. Still, similar to confidence score based techniques, EM-based algorithms might result in sub-optimal final models because every EM-based algorithm is subject to the initial parameter $\boldsymbol{\theta}^0$.

This sub-optimality problem can be tackled by applying some prior knowledge when creating $\mathcal{S}^t$. In particular, [78] used a closeness measure in a feature space $\mathcal{X}$ to select samples, and in [112], a performance-driven measure was used instead of confidence scores. Specifically, in [112], unlabeled samples were first divided into m groups, denoted as $G_1,\ldots,G_m$, such that the associated confidence scores in each group were distributed similarly. Next, performance-driven measures, defined similar to Eq. (3), were evaluated after training a candidate parameter vector $\boldsymbol{\theta}_j^{t+1}$ for the j^{th} group along with a labeled data set at time t, say $\mathcal{L}^t \cup G_j$, for $\forall j = 1\ldots m$. Then, a group, say G_k, was chosen for $\mathcal{S}^t$ if the largest performance gain was achieved with $\boldsymbol{\theta}_k^{t+1}$ against $\boldsymbol{\theta}^t$. In [112], it was shown that these procedures exhibited a consistent performance improvement over a confidence score based method. However, as one can imagine, the computational complexity of this method could be prohibitive as m different candidate parameter vectors (i.e., $\boldsymbol{\theta}_j^{t+1}$ for $\forall j = 1\ldots m$) need

to be trained at every iteration.

Finally, the Co-training technique discussed in Section 2.4 can also be considered a semi-supervised incremental learning technique, suitable for a case when there are two distinct feature spaces (e.g., $\mathcal{X} = (\mathcal{X}^{(1)}, \mathcal{X}^{(2)})$). The theoretical background of the technique is presented in Section 2.4 and referenced therein. Instead, we mention the underlying assumptions of the Co-training method here as follows: (a) each feature space should be *sufficient*, meaning that the use of a single feature is enough to achieve a perfect classification accuracy, and (b) each feature space needs to be conditionally independent given a class label. Note that satisfying both assumptions is sometimes unrealistic, so the Co-training algorithm might not perform any better than the confidence score based technique if the assumptions do not hold. We summarize the detailed procedures of the Co-training algorithm in Algorithm 2 for more consistent presentation of related work.

Algorithm 2 Co-training algorithm [13]

prepare $\mathcal{U}^0$ and $\mathcal{L}^0$
initialize $\boldsymbol{\theta}^{(1)0}$ using $\mathcal{L}^0$ with respect to $\mathcal{X}^{(1)}$
$t \leftarrow 0$
repeat
  for $j = 1, 2$ **do**
    compute confidence scores $s_c(\delta(x; \boldsymbol{\theta}^{(j)t}))$ for $x \in \mathcal{U}^t$ with respect to $\mathcal{X}^{(j)}$
    $N_u^t \leftarrow |\mathcal{U}^t|$
    $k_u^t \leftarrow$ the number of samples to be selected at time t
    $\tilde{\mathcal{U}}^t \leftarrow \{x_{(i)} \mid x_{(i)} \in \mathcal{U}^t,\ s_c(\delta(x_{(1)}; \boldsymbol{\theta}^{(j)t})) \geq s_c(\delta(x_{(2)}; \boldsymbol{\theta}^{(j)t})) \geq \cdots \geq s_c(\delta(x_{(N_u^t)}; \boldsymbol{\theta}^{(j)t}))\}$
    $\mathcal{S}^t \leftarrow \{(x_{(i)}, y_{(i)}) \mid i \leq k_u^t,\ y_{(i)} = \delta(x_{(i)}; \boldsymbol{\theta}^{(j)t}),\ x_{(i)} \in \tilde{\mathcal{U}}^t\}$
    $\mathcal{L}^t \leftarrow \mathcal{L}^t \cup \mathcal{S}^t$
    $\mathcal{U}^t \leftarrow \mathcal{U}^t \backslash \mathcal{S}^t$
    if $j = 1$ **then**
      update $\boldsymbol{\theta}^{(2)t+1}$ with $\mathcal{L}^t$
    else if $j = 2$ **then**
      update $\boldsymbol{\theta}^{(1)t+1}$ with $\mathcal{L}^t$
    end if
  end for
  if $\boldsymbol{\theta}^{(j)t+1} = \boldsymbol{\theta}^{(j)t}$ for $j = 1, 2$ **then**
    break
  else
    $t \leftarrow t + 1$
  end if
until $|\mathcal{U}^t| = 0$

2.4 Relevant multi-view learning techniques

In many pattern classification problems, using multiple features is very important for constructing a good concept model because a single feature is often too simple to capture the essence of signals. For example, in image concept modeling, the outputs of steering filters [40], SIFT [66], and Gabor wavelets [37] can be combined to improve modeling accuracy [81]. Intuitively, working with multiple features can be thought of as a process of integrating multiple cues, each of which is specialized to deal with a certain characteristic of training data. Because the cues from different features often exhibit different levels of discriminating power, a key question to ask is how to produce an unified classification result while considering the quality of each feature.

In supervised learning literature, there are mainly two different techniques: (a) early fusion, and (b) late fusion. In early fusion, features are combined before a model is learned, so only a single classifier is generated during training. One advantage of early fusion is that it is very easy to implement; one can simply put feature vectors next to each other, creating a single, large feature vector. However, for some heterogeneous features such as audio features and visual features, it is not straightforward to combine these features because given a video stream for example, audio is usually processed at every ten milliseconds, while video is refreshed at every one twenty-fourth second. Moreover, in early fusion, finding the right balance between features is not a trivial task.

On the other hand, in late fusion, the output scores from the classifiers trained on individual features, namely base classifiers, are combined by taking the scores as an input of a meta-classifier. In the literature, late fusion is sometimes referred to as model-based transform (MBT) [115, 51, 101]. An advantage of late fusion over early fusion is that the meta-classifier automatically learns weights between features depending on the discriminating power of each feature. An additional benefit of late fusion is that it is easy to conduct a post-analysis regarding how well each feature performs. However, it is somewhat difficult to apply the late fusion approach to a semi-supervised setting in that it tends to over-fit to training data, making the output scores of base classifiers bi-modal, which, limits the merit of having multiple features. In other words, there is no room for further improvement

through a meta-classifier. Therefore, in semi-supervised learning (SSL), early fusion has been a major technique to deal with multiple features.

In the SSL literature, training a classification model with multiple features is often referred to as *multi-view learning* [10, 29, 90]. The origin of multi-view learning can be traced back to the late '90s when A. Blum and T. Mitchell proposed their seminal work for multi-view semi-supervised learning, namely, the Co-training [13]. In their work, iterative procedures were proposed for two feature space cases (i.e., $\mathcal{X} = (\mathcal{X}^{(1)}, \mathcal{X}^{(2)})$), where a parameter vector for the first feature space $\mathcal{X}^{(1)}$, denoted as $\boldsymbol{\theta}^{(1)}$ was bootstrapped by a classification model of the second feature space $\mathcal{X}^{(2)}$, $\boldsymbol{\theta}^{(2)}$, and vice versa. To develop such procedures, they imposed a *compatible assumption* on both classification models, stating that if the marginal distribution of a sample x is non-zero, classification results from the first and the second views should be equal. Given this assumption, they proved that arbitrarily good classification performance could be obtained given the following two conditions: (a) conditional independence between $x^{(1)} \in \mathcal{X}^{(1)}$ and $x^{(2)} \in \mathcal{X}^{(2)}$ when a class label was given, and (b) sufficiency of $\mathcal{X}^{(1)}$ and $\mathcal{X}^{(2)}$. The sufficiency of a feature space was defined as follows: given enough number of training samples, one could obtain perfect prediction results. In [68], these conditions were empirically verified by constructing a multi-view environment artificially, splitting a single feature into two distinct features. However, as concluded in [68], satisfying all of the conditions imposed by the original Co-training algorithm is somewhat unrealistic. As a response, there has been a series of studies, such as [4, 105], to uncover more relaxed conditions than those in [13]. Among them, the results in [105] were particularly pleasing as it proved that the Co-training would succeed if classifiers trained on individual views were initially different. Nevertheless, all of these studies still assumed that each view had to be sufficient.

In the meantime, an analysis in [28] triggered another set of studies for multi-view learning. In [28], it was proved that a classification error would be upper-bounded by the probability of disagreement between views (i.e. $P(f^{(1)}(x, y; \boldsymbol{\theta}^{(1)}) \neq f^{(2)}(x, y; \boldsymbol{\theta}^{(2)})$, where $f^{(1)}$ and $f^{(2)}$ are discriminant functions for the feature space $\mathcal{X}^{(1)}$ and $\mathcal{X}^{(2)}$ parametrized with $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$, respectively). Starting from a model complexity perspective, the work

in [36, 63, 79] reached the same conclusion as the one in [28]. Specifically, it was studied, in [36, 63, 79], that minimizing the level of disagreement would reduce the Rademacher complexity, a measure of the complexity of a parameter space [5]. It is well-known if the complexity is reduced, the generalization error of a classification system on unseen data will also be decreased [58]. Later, based on the same principle as the one in [28], co-regularized SSL techniques were proposed in [90, 15, 92, 111], where the amount of disagreement between views was measured by a squared sum of differences between the values of discriminant functions $f^{(1)}$ and $f^{(2)}$ over an unlabeled data set $\mathcal{U}$ given by

$$\sum_{i=1}^{N_u} \sum_{y \in \mathcal{Y}} \left[f^{(1)}(x_{u_i}^{(1)}, y; \boldsymbol{\theta}^{(1)}) - f^{(2)}(x_{u_i}^{(2)}, y; \boldsymbol{\theta}^{(2)}) \right]^2. \tag{22}$$

It can be easily seen that Eq. (22) is convex with respect to the values of $f^{(1)}$ and $f^{(2)}$ for $x_{u_i} \in \mathcal{U}$. In fact, this convexity makes Eq. (22) possible to be incorporated into a regular discriminative learning framework easily. In particular, in [90, 15, 92, 111], Eq. (22) was used as an additional regularization term on top of the empirical error term and the regularization term in Eq. (3). Nevertheless, it has not been fully verified that the use of Eq. (22) is an optimal way to take advantage of multiple features in a semi-supervised setting. Thus, in Chapter 5, we investigate an agreement function for multi-view learning so that we can differentiate the level of agreement we want to impose on for each unlabeled sample.

Chapter III

KERNELIZED MAXIMAL-FIGURE-OF-MERIT (KMFOM) LEARNING FOR IMAGE CONCEPT MODELING

When evaluating the success of a machine learning algorithm, a certain performance metric is selected depending on the type of problem to which the algorithm is applied. In many cases however, there is inconsistency between the performance metric used in testing phases and that used in training phases. While this inconsistency does not always result in performance degradation, it is often seen that systems trained and tested consistently outperform systems learned and evaluated with the inconsistency. Thus, several techniques that optimize performance metrics directly have been recently proposed [16, 55, 43].

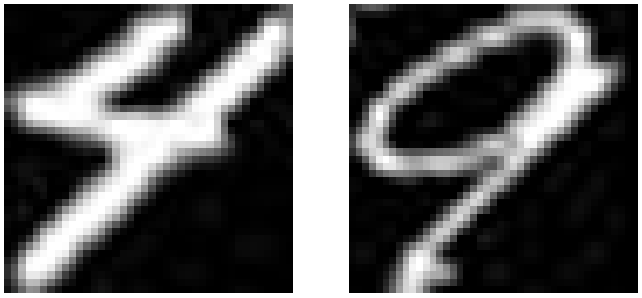
Among them, an MFoM learning approach has been successfully applied to many machine learning problems, such as text categorization [43], automatic image and video annotation [42, 21], and image spam detection [20]. The key idea was to define continuous and differentiable functions for a class dependent score function $g_y(x)$ for a class $y \in \mathcal{Y}$, where $\mathcal{Y}$ is a set of class labels, and to use an optimizable objective function that approximates commonly used performance metrics, such as a false positive error, a false negative error, or the ranking of a sample x. Although this MFoM learning approach is relatively easy to apply to a wide range of performance metrics, the effectiveness of this technique has been limited by the fact that so far, $g_y(x)$ can only be a linear function defined by

$$g_y(x) = \langle \theta_y, x \rangle_{\mathcal{X}} + \theta_{0y}, \tag{23}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{X}}$ is an inner product in $\mathcal{X} \subset \mathbb{R}^d$, and $\theta_{0y} \in \mathbb{R}$ is the intercept of a hyperplane in a d-dimensional space for the class y determined by $\theta_y = [\theta_{1y}, \ldots, \theta_{dy}]$.

Such a restriction also greatly affects the capability of the technique to model image concepts. In particular, the closeness between two image feature vectors does not linearly transfer to the similarity in image concepts in many cases. To see this, consider two images from the USPS handwritten digit recognition task data set in Figure 4. Although they

are corresponding to two different numbers, number four and number nine, they look very similar, making distinguishing these images using a linear model a difficult task.



(a) number four (b) number nine

Figure 4: Two images extracted from the USPS handwritten digit recognition task data set. Although they correspond to two different numbers, four and nine, respectively, their appearances are very similar to each other. Therefore, the closeness in image feature vector domains might not be linearly mapped into the similarity in the class label domain.

As briefly mentioned in Section 2.1.1, $g_y(x)$ in Eq. (23) can be non-linearized through a feature map Ψ on $\mathcal{X}$ given by $\Psi : \mathcal{X} \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space. In particular, given $\Psi(x)$, a nonlinear $g_y(x)$ can be written as a linear combination of a linear functional on $\mathcal{H}$ (i.e., an inner product in $\mathcal{H}$) as follows:

$$g_y(x) = \sum_{i=1}^{\infty} \theta_{iy} \langle \Psi(x_i), \Psi(x) \rangle_{\mathcal{H}} + \theta_{0y} \tag{24}$$

for all elements in $\mathcal{X}$ enumerated by $x_i \in \mathcal{X}$. However, one easily notices that $\theta_y = [\theta_{1y}, \ldots, \theta_{\infty y}]$ is now an infinite dimensional vector, which is impossible to determine. This difficulty can be addressed by *the Representer theorem*, stating that given a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined as $k(x', x) = \langle \Psi(x'), \Psi(x) \rangle_{\mathcal{H}}$ and the associated RKHS, $\mathcal{H}_K$, $g_y(x)$ has to be in the following form:

$$g_y(x) = \sum_{i=1}^{N} \alpha_{iy} k(x_i, x) + \alpha_{0y}, \tag{25}$$

where N is the total number of training samples, $\alpha_y = [\alpha_{1y}, \ldots, \alpha_{Ny}]$ is a dual parameter in $\mathbb{R}^N$, and α_{0y} is the intercept of a hyperplane determined by α_y. Thus, we only need to estimate a N-dimensional vector α_y. Now, one might attempt to derive a non-linearlized version of an MFoM learning approach by plugging-in Eq. (25) into the original MFoM learning framework. However, this simple approach might be problematic as α_y can still be a very high dimensional vector if the number of training data samples, N, is large.

In this chapter, we develop a kernelized MFoM learning approach using subspace distance minimization for image concept modeling. In particular, a subset of training samples whose cardinality is Q is selected in a way that a subspace distance between a subspace constructed with the entire training data set and that obtained from the subset is minimized. Intuitively, this subset-selection method can be considered as a process for finding a Q-dimensional function space that covers the original function space as much as possible. To this end, we exploit the definition of the subspace distance presented in [103]. Then, we show that this distance can be minimized by the Nyström extension, a spectral decomposition problem studied in [6]. An efficient algorithm to perform the Nyström extension is also proposed using a rank-1-update and -downdate algorithm given a Cholesky decomposition of a kernel matrix associated with the training data samples. We then verify the effectiveness of the proposed learning approach on three different image concept modeling problems, such as handwritten digit recognition, object recognition, and automatic image annotation.

This chapter is organized as follows. In Section 3.1, we present mathematical formulations to find the subset of training data that minimizes the subspace distance using the Nyström extension. In Section 3.2, the overall kernelized MFoM learning framework is presented, followed by a set of experimental results on various image concept modeling problems in Section 3.3. Finally, this chapter is concluded in Section 3.4 with some remarks on the related work. Note that the initial work was presented in [18].

3.1 Subspace distance minimization through the Nyström extension

To ease demonstration efforts, let us assume that there is an index set $\mathcal{I} = \{i : 1 \leq i \leq N\}$, where $i \in \mathcal{I}$ denotes the i^{th} sample in a training data set $\mathcal{D}$, and $\mathcal{I}_s = \{j : 1 \leq j \leq Q\}$ with the cardinality of Q, a subset of $\mathcal{I}$, where $j \in \mathcal{I}_s$ denotes the j^{th} sample in the subset. Then, using $\mathcal{I}$ and $\mathcal{I}_s$, the problem can be restated as follows: to find $\mathcal{I}_s$ such that the distance between a subspace constructed by $\mathcal{I}_s$ and a subspace constructed by $\mathcal{I}$ is minimized. To this end, let us define a distance between subspaces as presented in [103]:

Definition 1. [103] Let $\mathcal{S}$ be a linear space, and $\mathcal{U}$ and $\mathcal{V}$ be l- and m-dimensional subspaces of $\mathcal{S}$, respectively. If $u_1, \ldots, u_l$ and $v_i, \ldots, v_m$ are the orthonormal bases of $\mathcal{U}$ and $\mathcal{V}$, then a

distance between $\mathcal{U}$ and $\mathcal{V}$ can be defined as

$$d(\mathcal{U},\mathcal{V}) = \sqrt{\max(l,m) - \sum_{i=1}^{l}\sum_{j=1}^{m} \langle u_i, v_j\rangle_{\mathcal{S}}^2}, \tag{26}$$

where $\langle\cdot,\cdot\rangle_{\mathcal{S}}$ represents an inner product in the linear space $\mathcal{S}$.

From Eq. (26), $d(\mathcal{U},\mathcal{V})$ can be interpreted as a root-mean-square error of a projection of the basis vectors of $\mathcal{U}$ onto $\mathcal{V}$. Note that $d(\mathcal{U},\mathcal{V})$ achieves its maximum when the two spaces are orthogonal to each other, while it achieves its minimum when the two spaces are fully overlapped.

Now given Definition 1, let $\mathcal{U}$ be a subspace of a Hilbert space $\mathcal{H}$ given by a feature map $\Psi : \mathcal{X} \to \mathcal{H}$ with $\mathcal{I}$, and $\mathcal{V}$ be a subspace of $\mathcal{H}$ constructed $\mathcal{I}_s$. In particular, they can be written as follows:

$$\mathcal{U} = span\{\Psi(x_i)|i \in \mathcal{I}\} \tag{27}$$

$$\mathcal{V} = span\{\Psi(x_j)|j \in \mathcal{I}_s\}. \tag{28}$$

One difficulty of using Eq. (27) and Eq. (28) to compute Eq. (26) is that the orthonormal basis vectors for $\mathcal{U}$ and $\mathcal{V}$ are unknown. We address this difficulty by transforming a problem of minimizing Eq. (26) into a spectral decomposition problem using Lemma 2. To be more specific, suppose there are kernel matrices K and K_s, where $\{K\}_{ij} = k(x_i, x_j)$ and $\{K_s\}_{i'j'} = k(x_{i'}, x_{j'})$ for $\forall i, j \in \mathcal{I}$ and $\forall i', j' \in \mathcal{I}_s$. Suppose further that the rows and the columns in K have been rearranged so that the first Q rows and the first Q columns correspond to K_s. Then, block matrices A, B, and K_s, K can be written as follows:

$$K = \begin{bmatrix} K_s & A \\ A^T & B \end{bmatrix}, \tag{29}$$

where the rows and the columns of $A \in \mathbb{R}^{Q\times N-Q}$ correspond to $\mathcal{I}_s$ and $\mathcal{I}_s^c$, a complementary set of $\mathcal{I}_s$ given by $\mathcal{I}_s^c = \mathcal{I}\backslash\mathcal{I}_s$, respectively. Similarly, the rows and the columns of $B \in \mathbb{R}^{N-Q\times N-Q}$ are all associated with $\mathcal{I}_s^c$. Whence, the following lemma holds:

Lemma 2. *let σ_i be an eigenvector of K in Eq. (29) with λ_i to be σ_i's associated eigenvalue. Furthermore, let $u_1, \ldots, u_l$ and $v_i, \ldots, v_m$ be the orthonormal basis vectors of $\mathcal{U}$ and $\mathcal{V}$, a*

subspace of $\mathcal{S}$ respectively. Then the following equality holds:

$$\sum_{j=1}^{m} \langle u_i, v_j \rangle_{\mathcal{S}}^2 = \frac{1}{\lambda_i} \sigma_i^T \tilde{K} \sigma_i, \tag{30}$$

where

$$\tilde{K} = \begin{bmatrix} K_s & A \\ A^T & A^T K_s^{-1} A \end{bmatrix}. \tag{31}$$

The proof of Lemma 2 is presented in the Appendix.

Lemma 2 shows that the squared sum of the inner products between the orthonormal basis vectors of $\mathcal{U}$ and $\mathcal{V}$ represented in Eq. (26) can be written as a quadratic form of a matrix $\tilde{K}$ with an eigenvector σ_i of K. Now, given Lemma 2 one can derive an upper bound of Eq. (26) as follows:

$$\begin{aligned} d(\mathcal{U}, \mathcal{V}) &= \sqrt{\max(l, m) - \sum_{i=1}^{l} \sum_{j=1}^{m} \langle u_i, v_j \rangle_{\mathcal{H}}^2} && (32) \\ &= \sqrt{\max(l, m) - \sum_{i=1}^{l} \frac{1}{\lambda_i} \sigma_i^T \tilde{K} \sigma_i} && (33) \\ &= \sqrt{\sum_{i=1}^{l} \frac{1}{\lambda_i} \sigma_i^T K \sigma_i - \sum_{i=1}^{l} \frac{1}{\lambda_i} \sigma_i^T \tilde{K} \sigma_i} && (34) \\ &\leq \sqrt{\sum_{i=1}^{l} \frac{1}{\lambda_i}} \left\| K - \tilde{K} \right\|_F, && (35) \end{aligned}$$

where $\| \cdot \|_F$ is a Frobenius norm of a certain matrix. From Eq. (33) to Eq. (34), the fact that $\frac{1}{\lambda_i} \sigma_i^T K \sigma_i = 1$ is used. Moreover, without the loss of generality, we assume that $l \geq m$. Now, since Eq. (35) holds for any $\mathcal{U}$ and $\mathcal{V}$,

$$\min_{\mathcal{I}_s \subset \mathcal{I}} d(\mathcal{U}, \mathcal{V}) \leq \varsigma \min_{\mathcal{I}_s \subset \mathcal{I}} \left\| K - \tilde{K} \right\|_F, \tag{36}$$

where $\varsigma = \sqrt{\sum_{i=1}^{l} \frac{1}{\lambda_i}}$, is also true. Based on Eq. (36), one can see that instead of minimizing $d(\mathcal{U}, \mathcal{V})$ directly, we can minimize the right-hand side of Eq. (36), the Frobenius norm of the difference between the original kernel matrix K and its approximate, $\tilde{K}$, to find $\mathcal{I}_s$.

In fact, the minimization of the right-hand side in Eq. (36) is a certain spectral decomposition problem, known as the Nyström extension [106, 39, 6]. To solve the Nyström

extension, we adopt a technique proposed in [6]. The basic idea of [6] is to sample $\mathcal{I}_s$ according to a probability distribution that is proportional to the determinant of K_s. To implement such a sampling strategy, in [6], a Metropolis-Hastings algorithm [9] is used, where at every iteration, an element in $\mathcal{I}_s$ is swapped with an element in $\mathcal{I}_s^c$. A proposal statistic, defined by a ratio of the determinant of K_s before swapping to the determinant of K_s after swapping, is then used to determine whether to accept the swapped element or not.

One problem of the use of the Metropolis-Hastings algorithm is that computing the determinant of a Q-by-Q matrix requires computational complexity of $\mathcal{O}(Q^3)$. To address this problem, in our kernelized MFoM learning framework, we propose a rank-1-update and -downdate algorithm to evaluate the determinant of K_s. In particular, given a lower triangular matrix obtained from the Cholesky decomposition of K_s, we show that the swapping operation can be done by a rank-1-update followed by a rank-1-downdate of the lower triangular matrix. Since the rank-1-update and the rank-1-downdate of a lower triangular matrix can be done in $\mathcal{O}(Q^2)$ as shown in [44], computation of the proposal statistic becomes very efficient. The detailed derivation of using a rank-1-update and -downdate algorithm for the swapping operation is presented the Appendix.

3.2 The proposed kMFoM learning framework

Once the subset of training data $\mathcal{I}_s$ is found, the class score function g_y, initially defined in Eq. (25) for a class y in $\mathcal{Y}$, where $\mathcal{Y}$ is a set of class labels given by $\mathcal{Y} = \{1, \ldots, C\}$, can be rewritten as

$$g_y(x) = \sum_{j=1}^{Q} \alpha_{jy} k(x_j, x) + \alpha_{0y}, \tag{37}$$

where x_j represents the j^{th} sample in the $\mathcal{I}_s$. Given this score function, an objective function to learn the dual parameter vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_C]$, where α_y is a $Q+1$-dimensional parameter vector of the class y, is derived from Eq. (3). In particular, we set $R(f(\cdot;\boldsymbol{\theta}))$, a regularization term, to a sum of squared L_2-norms of the class score functions, g_y's, for $y \in \mathcal{Y}$ restricted to the subspace $\mathcal{V}$ (recall that in Eq. (1), a discriminant function f is defined as a difference between g_y and g_{y^-} for a class y and $y^- = \mathcal{Y} \backslash y$.). Thus, the objective

function is written as

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^{C\times(Q+1)}} \sum_{i=1}^{N} V(f(x_i,1;\boldsymbol{\alpha}),\ldots,f(x_i,C;\boldsymbol{\alpha}),y_{l_i};\mathcal{D}) + \lambda \sum_{y\in\mathcal{Y}} \|g_y\|_{\mathcal{V}}^2, \tag{38}$$

where $\|g_y\|_{\mathcal{V}}^2$ is a squared L_2-norm of g_y evaluated on $\mathcal{V}$, λ is a positive parameter to control the balance between the L_2-norm and the loss function V. Note that V in Eq. (38) is written as a function of discriminant functions for all classes because some performance metrics, such as micro- or macro-averaging F_1-measures, are dependent upon all of the quantities. Similarly, in an MFoM learning framework, V can be defined in several different ways depending on a performance metric of interest. For example, suppose that we want to minimize a classification error rate. The loss function V is then defined as

$$V(\cdot;\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} l(x_i,y_i;\boldsymbol{\alpha}), \tag{39}$$

where $l(x_i,y_i;\boldsymbol{\alpha})$ is a class 0-1 loss function for the i^{th} sample in $\mathcal{D}$, x_i, with its class label y_i while N is the total number of training samples in $\mathcal{D}$. The class 0-1 loss function $l(x_i,y_i;\boldsymbol{\alpha})$ in Eq. (39) is typically defined using a sigmoid function given by

$$l(x_i,y_i;\boldsymbol{\alpha}) = \frac{1}{1+e^{\alpha f(x_i,y_i;\boldsymbol{\alpha})+\beta}}, \tag{40}$$

where α and β are constants to determine the slope and the offset of the sigmoid function. Note that Eq. (40) becomes unity (or zero) when the value of a discriminant function is negative (or positive), which is consistent with the definition of a classification error because according to the decision rule in Eq. (2), an error occurs whenever $f(x_i,y_i;\boldsymbol{\alpha}) < 0$.

Another example is an F_1-measure. As shown in Table 1, the F_1-measure is a harmonic mean of recall and precision (refer to Table 1 for the definitions of recall and precision as well). Assuming there are only two classes, say positive and negative, denoted as c_+ and c_-, the loss function V for the F_1-measure can be written as a function of the number of true positive samples, a false positive error, and a false negative error, each of which is denoted as TP, FP, and FN, respectively, as follows:

$$V(\cdot;\mathcal{D}) = 1 - \frac{2TP}{2TP+FN+FP}, \tag{41}$$

where the terms of TP, FP, and FN in Eq. (41) are given by

$$TP = \sum_{i=1}^{N} (1 - l(x_i, y_i; \boldsymbol{\alpha})) I(y_i = c_+), \tag{42}$$

$$FP = \sum_{i=1}^{N} (1 - l(x_i, y_i; \boldsymbol{\alpha})) I(y_i \neq c_+), \tag{43}$$

$$FN = \sum_{i=1}^{N} l(x_i, y_i; \boldsymbol{\alpha}) I(y_i = c_+), \tag{44}$$

where $I(\cdot)$ is an indicator function and $l(\cdot; \boldsymbol{\alpha})$ is a class 0-1 loss function defined in Eq. (40). In essence, the loss function V in Eq. (41) is maximizing the F_1-measure for the class c_+. Additional performance metrics, such as precision, recall, or average precision can be derived in a similar fashion.

3.3 Experimental results on image concept modeling

For evaluation purposes, we tackled three different image concept modeling problems: handwritten digit recognition using the USPS data set, object recognition using the COIL-100 data set, and image annotation with the Corel 5k data set. In Figure 5, we illustrate some of the handwritten digit images extracted from the USPS data set, which demonstrate a wide range of writing styles in the data set. The images in this data set were collected from a database of addresses and ZIP codes gathered at the Buffalo Post Office, New York, USA, creating a collection of 18468 images corresponding to digits from zero to nine. Out of the 18468 images, we randomly drew 150 images for each digit and labeled the images corresponding to digits, two and five, as positive and the rest as negative. We then down-sampled the images to 16-by-16 and randomly picked 20% for evaluation, resulting in a split of 1200 images for training and 300 images for testing. As for the feature vectors, we used pixel values directly with which an RBF kernel was computed on top of the feature vectors, where the bandwidth parameter h of the kernel was simply set to the average Euclidean distance of the feature vectors.

Figure 6 depicts some sample images from the COIL-100 data set, which consists of images of a set of 100 objects captured from 72 different angles, rotating at every five degrees. As shown in Figure 6, the objects exhibit quite a bit of distinct geometric and reflectance

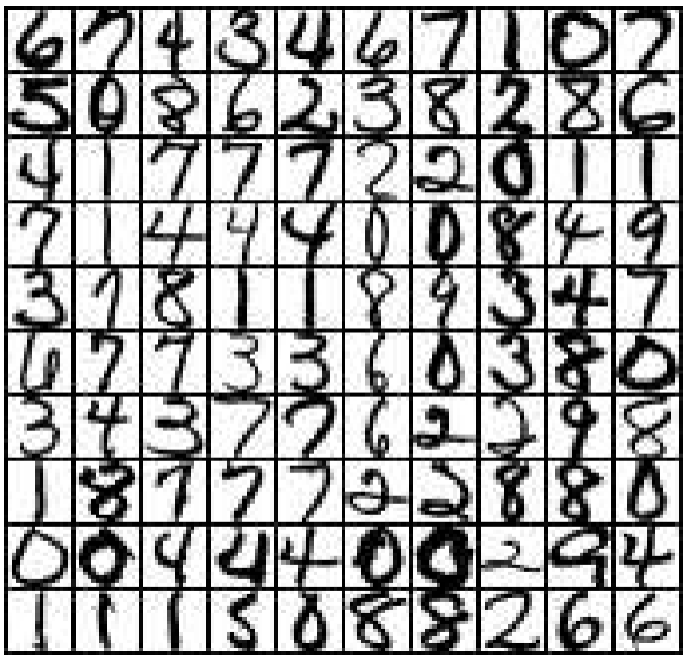


Figure 5: Examples of handwritten digit images extracted from the USPS data set

characteristics. To evaluate the proposed system on this data set, we first randomly selected 24 objects out of 100, resulting in 1728 images. Similar to the USPS data set, we then performed down-sampling of the images into 16-by-16. The set of 24 objects was partitioned into six classes of four object each. Moreover, randomly chosen 38 images from each class were discarded to leave 250 image each. Finally, we split the remaining 1500 images into two sets; one consisting of 1200 images for training and the other consisting of 300 images for testing. As for the feature vectors, we again exploited pixel values directly to compute a kernel matrix where the kernel matrix was computed using the same algorithm as the one used for the USPS data set.

Finally, in Figure 7, some of the images contained in the Corel 5k data set and their associated semantic concepts are shown. Note that since image annotation is a multi-label problem, multiple concepts are allowed to be labeled to a single image. In the Corel 5k data set, there were 4500 training images and 500 test images with 374 semantic concepts, such as *city*, *mountain*, *hills*, *Boeing*, *flower*, *sky*, *tree*, *castle*, and many more (please refer to [72] for a complete list of the concepts.). Out of the set of 374 concepts, we selected 36 semantic concepts in which at least 100 training samples existed for benchmarking, resulting in 4212

Figure 6: Some object images extracted from the COIL-100 data set

training images and 467 test images in total. Feature vectors were extracted through the following steps. First, each image in the data set was segmented into 16-by-16 regular grids. From each grid, a 12-dimensional color feature vectors was extracted by computing the mean and the variance of each of the color channels in an RGB color space and a Lab color space, respectively. Because the resolution of the images were all 192-by-128, the number of feature vectors generated from each image was 96. Given feature vectors, clustering (e.g., k-means clustering, etc) was then performed to create a so-called visual lexicon, where the centroid of each cluster was assigned to a visual word. This visual lexicon was then used to index the grids generated earlier in the feature extraction process by associating the feature vector extracted from each grid with the closest centroid (i.e., a visual word). Then, visual unigrams and bigrams were counted from each image. Figure 8 illustrates the corresponding procedures. In particular, the visual unigrams are simply computed as the number of occurrences of each visual word for each image. As for the visual bigrams, let v_{ij} be a visual word assigned to the grid at the i^{th} row and j^{th} column. Then, they are obtained by counting the number of co-occurrences of adjacent visual words of v_{ij} in eight directions and v_{ij}, such as $(v_{i-1,j-1}, v_{ij})$, $(v_{i-1,j}, v_{ij})$,..., $(v_{i+1,j+1}, v_{ij})$. Because the number of visual words was chosen to be 64 in this work, the dimensions of the resulting unigram

(a) Castle, Tree, Sky

(b) Cliff, Sea, Sky, Shore

(c) Flower, Garden, Tree

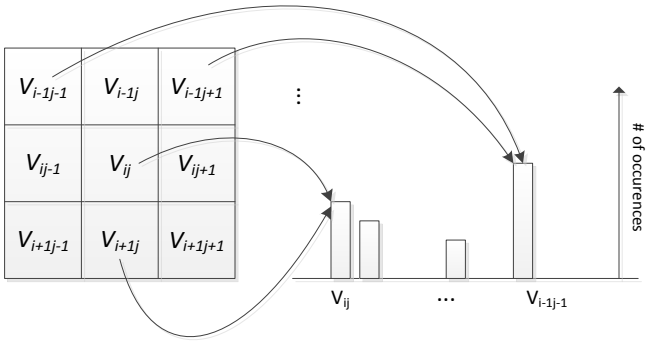
Figure 7: Some images from the Corel 5k data set and their associated semantic concepts

and bigram vector were 64 and 4096, respectively. The feature vectors for categorizing semantic concepts were then created by concatenating these two vectors (resulting in a 4160-dimensional vector). Given the feature vectors, We then performed a latent semantic indexing (LSI) [8] (a) to select visual unigrams or bigrams that had more discriminating power, and (b) to reduce the dimension of feature vectors for computational efficiency. The dimension of the final feature vectors was 600. Given these feature vectors, a kernel matrix was evaluated with an RBF kernel where a cosine distance $d_c(x_i, x_j)$ between two feature vectors x_i and x_j given by

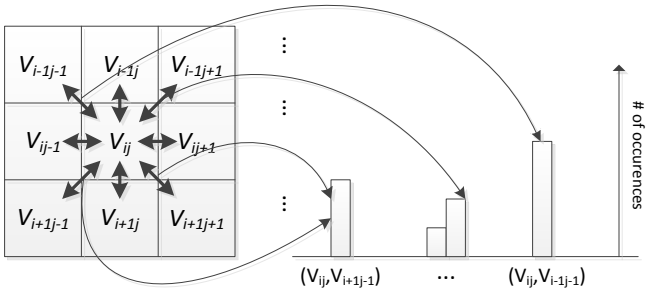
$$d_c(x_i, x_j) = 1 - \frac{x_i^T x_j}{\|x_i\|_2 \|x_j\|_2} \tag{45}$$

was used, where $\|\cdot\|_2$ represented an L_2-norm.

For the USPS data set and the COIL-100 data set, a classification error rate was selected as the preferred performance metric and macro-averaging F_1 was chosen as the performance metric for the Corel 5k data set. Macro-averaging F_1 is an average of F_1-measure over



(a) an illustration of visual unigram computation



(b) an illustration of visual bigram computation

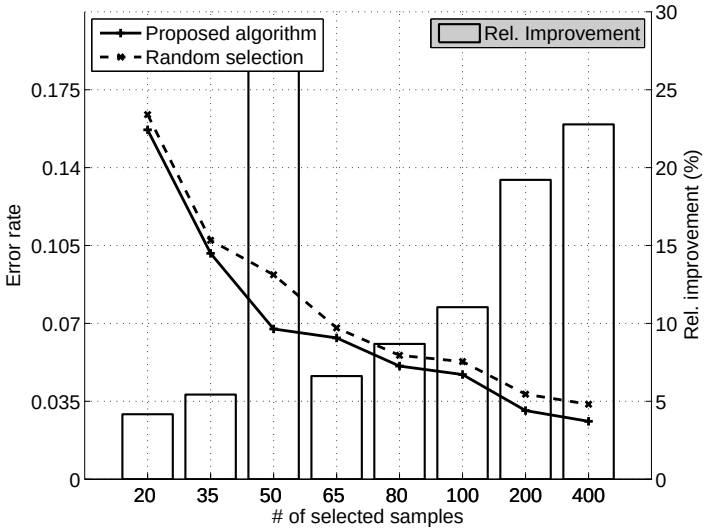
Figure 8: An illustration of the procedures to compute visual unigrams and bigrams. v_{ij} denotes a visual word assigned to the grid at the i^{th} row and the j^{th} column. In (a), visual unigrams are simply computed by counting the number of occurences of each visual word. Note that the visual words v_{ij} and v_{i+1j}, v_{i-1j-1} and v_{i-1j+1} are assumed to be the same. In (b), Given v_{ij}, we consider pairwise relationships of the visual words for neighboring grids in eight different directions. Each pair is then counted, generating a histogram of co-occurences of visual words. Note that the pairs (v_{ij}, v_{i+1j-1}) and (v_{ij}, v_{i+1j+1}) are assumed to be the same.

different classes without considering the number of samples belonging to the individual classes. Throughout the experiments, results were obtained after taking an average of 20 runs. All other parameters were determined empirically through cross-validation.

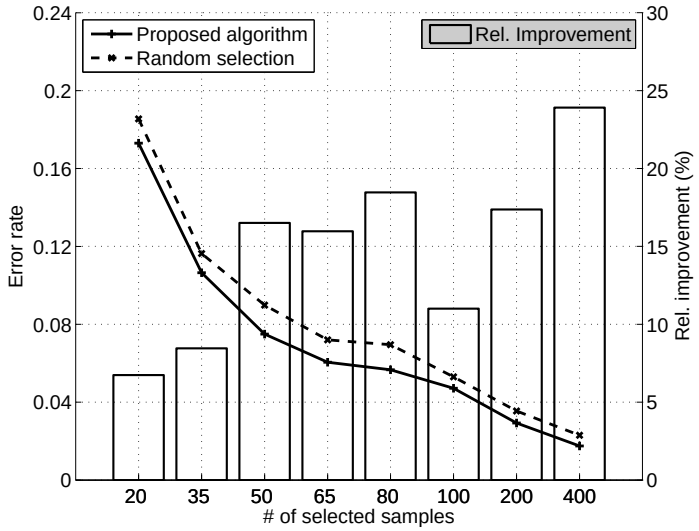
We set the baseline system as the same kernel MFoM learning framework but the subspace distance minimization algorithm was replaced with random selection. For fair comparison, all experimental configurations were set to be the same. We present the overall experimental results in Figure 9.

In Figure 9, varying the size of the subset Q (i.e., 20, 35, 50, 65, 80, 100, 200, 400), performance metrics (e.g., an error rate for the USPS and the COIL-100 data sets, and

an F_1-measure for the Corel 5k data set) of the proposed approach are compared with those of the baseline system. In Figure 9-(a) and -(b), it can be clearly seen that the proposed algorithm outperforms the baseline system in all cases. The maximum relative error rate reductions are 23.9% at $Q = 400$ and 26.5% at $Q = 50$ for the COIL-100 and USPS data sets, respectively. It is also seen that the amount of the relative improvement

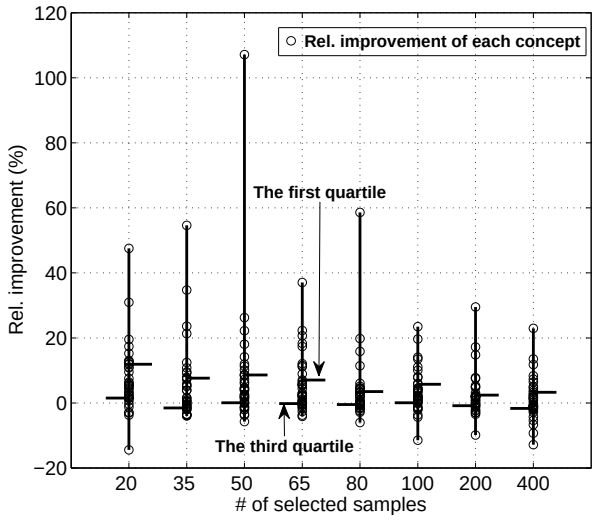


(a) USPS



(b) COIL-100

Figure 9
continued..



(c) Corel 5k

Figure 9 continued: Performance comparison graphs between the proposed kernelized MFoM learning approach and the baseline system, varying the size of the selection set, Q. In (a)-USPS and (b)-COIL-100, the primary axis(left) represents an error rate and the secondary axis(right) is the relative improvement. On the other hand, in (c)-Corel 5k, distributions of relative improvement are shown over different sizes of Q. Here, each circle corresponds to the relative improvement of the proposed technique over the baseline system for each concept. The end points of the vertical lines are the maximum and the minimum amounts of the relative improvement and horizontal lines represent the first and the third quartiles.

tends to increase as the training set size Q grows. For the Corel 5k data set, we present distribution graphs of the relative improvement for each concept as follows. In Figure 9-(c), each circle corresponds to each concept, resulting in 36 circles in each vertical line. The end points of each vertical line in Figure 9-(c) represent the maximum and the minimum relative improvement in term of an F_1 measure. The horizontal line on the right side of the vertical line is the first quartile, and the one on the left is the third quartile of the relative improvement of the proposed technique compared to the baseline system. Looking at the horizontal line indicating the third quartile, it can be clearly seen that for 75% out of the subset of 36 concepts, performance enhancement is observed. Moreover, for a quarter of the concepts, significant improvements can be seen. In fact, the amount of performance improvement is much larger than the amount of performance degradation for all cases. The most effective concept is *statue* followed by *valley* in which the amounts of relative improvement are 23.3% and 19.9% on average, respectively. The least effective concept is

jet where performance is relatively decreased by 1.4%. However, this amount is much less than the amounts of improvement achieved for other concepts, as discussed earlier.

The second set of comparison results are the time required for training vs. performances for the Corel 5k image data set. In Table 2, we list performances and training times when $Q = 400$ and $Q = 4212$. In Table 2, a clear advantage of the proposed technique is shown. In particular, it can be clearly seen that training time for the proposed technique is reduced by a factor of 30, while the performance remains comparable. This result demonstrates that the proposed framework is indeed an efficient algorithm to train a kernelized MFoM classifier.

Table 2: Comparisons on performances and training times while varying the size of the subset of training data

k=400		k=4212	
Macro F_1	Training time(s)	Macro F_1	Training time(s)
0.4346	162	0.4509	5031

3.4 Summary

In this chapter, we have presented a kernelized MFoM learning approach based on a subspace distance minimization criterion. We have provided an efficient algorithm to select a subset of training samples that leads to nearly optimal parameter estimation. Experimental results on several image data sets clearly showed that the proposed technique is efficient and effective to train nonlinear image concept models tailored to various performance metrics.

Besides the proposed kernelized MFoM learning approach, there are other efficient algorithms to learn kernelized classifiers. However, such algorithms are typically specialized for a certain loss function with or without a regularization term. In particular, a sequential minimal optimization (SMO) algorithm proposed in [73] for SVMs used the non-differentiability of a hinge-loss function. On the other hand, when the regularization term is set to an L_1-norm as in [96], training samples were first divided into a working set and a non-working set, and then a projected gradient-descent algorithm was exploited [83].

In spirit, our proposed algorithm shares the idea of isolating a working set from a non-working set as in [83]. However, it is different in essence from [83] in that a working set

is determined only once and the parameters for a non-working set are set to zero. This reduces the required computational complexity to at most $\mathcal{O}(NQ)$ (the exact computational complexity varies depending on an optimization algorithm used. For instance, a stochastic quasi-Newton method needs the complexity of $\mathcal{O}(Q^2)$.) to compute the gradient. Although one might argue that fixing a working set might result in a sub-optimal parameter vector, it should be noted that if the working set is not fixed, the computational complexity will remain high. Moreover, by carefully building a working set (e.g., the subspace constructed with the working set is close to the subspace obtained from the entire training data set), we will have a nearly optimal parameter vector at the end. It is also noteworthy that [106, 39] used the Nyström extension to train kernelized classifiers as well, but they purely aimed at approximating a kernel matrix and none of these approaches provided the relationship between a subspace distance and the Nyström extension as in Eq. (36).

Chapter IV

SEMI-SUPERVISED INCREMENTAL LEARNING WITH AN ERROR REDUCTION FUNCTION

In many machine learning problems, it is likely that designers are initially given only a small number of labeled samples accompanied by a large amount of unlabeled samples. Semi-supervised incremental learning is an attractive approach to bootstrapping a learning process, in this case by starting with an initial set of models learned with labeled samples, and then improving the initial models using unlabeled data.

To develop a semi-supervised incremental learning algorithm, a key question that needs to be addressed is which unlabeled samples should be incorporated. On one extreme, we can select samples that an existing model is confident about the associated class labels using a confidence score. However, as pointed out earlier in Chapter 2, this confidence score based approach might not always be beneficial in terms of improving the existing model because such samples are usually too similar to previously used samples. On the other extreme, motivated from *active learning*, we can select samples located near decision boundaries based on the fact that these samples could contribute the most to enhancing discrimination capabilities of an existing model. However, unlike *active learning* where an oracle exists (see Section 2.3.1 for more information), such samples are likely to be labeled incorrectly, so the incremental learning process might not converge.

As a compromise between these two extremes, we initially explored a combination of a confidence score and a margin-like discrimination score in [17]. Specifically, given an MFoM classifier at time t, we evaluated a confidence score based on a probabilistic approximation of the output of the MFoM classifier. We also measure the discrimination capability of a certain unlabeled sample x, called a margin-like discrimination score, using a regularized spectral clustering-based nearest neighbor (NN) classifier. These two scores were then simply added together and the resulting quantity was used to decide which unlabeled samples should be

selected. Experimental results were promising because a significant performance gain was observed when compared to the cases where the two above-mentioned scores were considered separately.

One interesting property of a margin-like discrimination score is that the closer a sample is to decision boundaries, the higher a margin-like discrimination score will be [17]. This implies that the score can be considered as a penalty term of a confidence score. Motivated by this observation, we investigate a more general semi-supervised incremental learning approach in this chapter. In particular, instead of using a margin-like discrimination score, we compute an expected error reduction function using a Bayesian decision theory to exactly quantify the amount of contribution that each unlabeled sample has in terms of reducing the overall classification error. Moreover, we generalize types of classifiers and the number of classifiers. Note that in [17], a linear MFoM classifier and a regularized spectral clustering-based NN classifier were used for sample selection. However, as discussed in Chapter 3, nonlinear classifiers might need to be exploited for image concept modeling to reduce modeling errors. On the other hand, the number of classifiers should not be limited only to two to increase the robustness in computing the amount of the contribution. Therefore, in this work, (a) instead of restricting ourselves to a *linear* MFoM classifier, a *kernelized* MFoM (kMFoM) classifier is used, and (b) more than two classifiers (e.g., a combination of three kMFoM and one spectral clustering-based NN classifier, a combination of eight kMFoM classifiers, etc.) can now be used to choose unlabeled samples. We refer to a set of such unlabeled samples as a selection set, denoted as $\mathcal{S}^t$ where the superscript t represents a discrete time index, incremented at every time when parameters are updated. The final touches of the proposed framework over the preliminary work are the uses of a Zipf distribution and a class prior probability distribution with which a potential class imbalance problem can be mitigated. We demonstrate the effectiveness of the proposed framework on two different image concept modeling problems, handwritten digit recognition with the USPS data set and object recognition with the COIL-100 data set, compared to two baseline systems, such as a confidence score based technique and a Co-training method, discussed in Section 2.3.2.

The remainder of this chapter is organized as follows. Section 4.1 describes our method

to calculate confidence scores followed by the algorithm to compute expected error reduction in Section 4.2. In Section 4.3, a technique using an ensemble of classifiers to improve the robustness of the expected error reduction to the variability and the bias caused by a small sample size of labeled data sets is discussed. The overall algorithm is presented in Section 4.4 and Section 4.5 discusses experimental results. Finally, Section 4.6 concludes this chapter.

4.1 A confidence score

A confidence score, denoted as $s_c(\delta(x_i;\boldsymbol{\theta}^t))$, is defined as the degree that a classification system, determined by a parameter vector $\boldsymbol{\theta}^t$ at time t, is confident in its decision $\delta(x_i;\boldsymbol{\theta}^t)$, given by Eq. (2), for the i^{th} sample x_i in the training data $\mathcal{D}$. The immediate implication of this definition is the fact that as the confidence score increases, the probability that the decision $\delta(x_i;\boldsymbol{\theta}^t)$ is incorrect will be reduced. As a result, the use of a confidence score can be typically seen in a verification system, where the system rejects its decision if a confidence score is less than a certain threshold. As for mathematical formulation of a confidence score, [53] provides a good literature survey on how to evaluate the confidence score. One popular choice is to use a generalized log-likelihood ratio (GLLR) defined as

$$GLLR(x_i, y_i;\boldsymbol{\theta}^t) = \log \frac{P(x_i|y_i;\boldsymbol{\theta}^t)}{\max_{y^- \in \mathcal{Y}\setminus y_i,} P(x_i|y^-;\boldsymbol{\theta}^t)}, \tag{46}$$

where $P(x_i|y_i;\boldsymbol{\theta}^t)$ is a class conditional probability of a sample x_i given its class label y_i parametrized with a vector $\boldsymbol{\theta}^t$. Note that as in the previous chapters, a total number of C classes are assumed, and $\boldsymbol{\theta}^t = [\theta_1^t, \ldots, \theta_C^t]$, where θ_j^t represents the parameter vector for the j^{th} class. Given Eq. (46), the confidence score for a decision $\delta(x_i;\boldsymbol{\theta}^t)$ is then given by

$$s_{conf}(\delta(x_i;\boldsymbol{\theta}^t))_{GLLR} = \log \frac{P(x_i|y;\boldsymbol{\theta}^t)}{\max_{y^- \in \mathcal{Y}\setminus y,} P(x_i|y^-;\boldsymbol{\theta}^t)}|_{y=\delta(x_i;\boldsymbol{\theta}^t)}, \tag{47}$$

where we use y without the subscript i for a class label variable to differentiate it from the ground-truth class label y_i for a sample x_i. One difficulty of handling Eq. (47) is the fact that its range is $(-\infty, \infty)$. Often, such a wide range of Eq. (47) causes problems when the actual value is used for further processing. To tackle this difficulty, in this work, we define the confidence score $s_{conf}(\delta(x_i;\boldsymbol{\theta}^t))$ for a sample x_i as follows:

$$s_{conf}(\delta(x_i;\boldsymbol{\theta}^t)) = \frac{P(x_i|y;\boldsymbol{\theta}^t)}{P(x_i|y;\boldsymbol{\theta}^t) + \max_{y^- \in \mathcal{Y}\setminus y} P(x_i|y^-;\boldsymbol{\theta}^t)}|_{y=\delta(x_i;\boldsymbol{\theta}^t)}, \tag{48}$$

so that now, the range of the confidence score becomes within the interval of $[0, 1]$. Because of this range characteristic, one might consider our confidence score $s_{conf}(\delta(x_i; \boldsymbol{\theta}^t))$ as a variant of a class posterior probability $P(y|x_i, \boldsymbol{\theta}^t)$ for a sample x_i. In fact, it is true to some extent that Eq. (48) is the same as the class posterior probability $P(y|x, \boldsymbol{\theta}^t)$ if we assume that there are only two classes and the prior distribution for each class is *uninformative prior*. Interestingly, this equivalence reveals another possible formulation for a confidence score based on the class posterior probability $P(y|x, \boldsymbol{\theta}^t)$, which we denote it as $s_{conf}(\delta(x; \boldsymbol{\theta}^t))_{POST}$ as follows:

$$s_{conf}(\delta(x_i; \boldsymbol{\theta}^t))_{POST} = \frac{P(x_i|y; \boldsymbol{\theta}^t)}{\sum_{y' \in \mathcal{Y}} P(x_i|y'; \boldsymbol{\theta}^t)}|_{y=\delta(x_i;\boldsymbol{\theta}^t)}. \tag{49}$$

If the number of classes is more than two, however, Eq. (47) starts to differ from Eq. (49) because of the maximization term in the denominator in Eq. (48). The use of this maximization term is to ensure Eq. (48) to be greater than 0.5 whenever the classification result given by $\delta(x_i; \boldsymbol{\theta}^t)$ is correct. In contrast, when there are more than two classes, say four for example, $s_c(\delta(x_i; \boldsymbol{\theta}^t))_{POST}$ might be much less than 0.5 even when the decision rule $\delta(x_i; \boldsymbol{\theta}^t)$ provides a true prediction output. This randomness might, in turn, pose a difficulty to use the score for determining whether to include a sample x into the incremental learning process consistently.

For discriminative learning algorithms discussed in Section 2.1, including a kernelized MFoM learning approach presented in Chapter 3, the class conditional probability $P(x_i|y; \boldsymbol{\theta}^t)$ for $y \in \mathcal{Y}$ used in Eqs. (47)-(49) is not given. In this case, a technique that generates a probabilistic output of the discriminant function $f(x_i, y; \boldsymbol{\theta}^t)$ for a sample x_i and a class $y \in \mathcal{Y}$ can be exploited as proposed in [74]. More precisely, suppose the discriminant function $f(x_i, y; \boldsymbol{\theta}^t)$ is given by

$$f(x_i, y; \boldsymbol{\theta}^t) = g_y(x_i) - \max_{y^- \in \mathcal{Y} \setminus y} g_{y^-}(x_i), \tag{50}$$

as in Eq. (1). Then, a confidence score $s_{conf}(\delta(x_i; \boldsymbol{\theta}^t))$ is defined as

$$s_{conf}(\delta(x_i; \boldsymbol{\theta}^t)) = \frac{1}{1 + e^{-\alpha_y f(x_i, y; \boldsymbol{\theta}^t) + \beta_y}}|_{y=\delta(x_i;\boldsymbol{\theta}^t)}, \tag{51}$$

using a sigmoid function with parameters α_y and β_y that determines the slope and the offset of the function for class y, respectively. These parameters are estimated based on a Maximum Likelihood (ML) criterion over a labeled data set available at time t, denoted as $\mathcal{L}^t$. More precisely, the objective function is written as

$$\max_{\alpha_y,\beta_y\in\mathbb{R}} \sum_{i=1}^{N_l^t} t_y^t \log(\frac{1}{1+e^{-\alpha_y f(x_{l_i},y;\boldsymbol{\theta}^t)+\beta_y}})I(y_{l_i}=y)+(1-t_y^t)\log(\frac{e^{-\alpha_y f(x_{l_i},y;\boldsymbol{\theta}^t)+\beta_y}}{1+e^{-\alpha_y f(x_{l_i},y;\boldsymbol{\theta}^t)+\beta_y}})I(y_{l_i}\neq y), \tag{52}$$

where the subscript l_i is used to denote the i^{th} labeled sample in $\mathcal{L}^t$ (note that the subscript i used so far represents the i^{th} sample in the total training data $\mathcal{D}$.), N_l^t is the total number of labeled data samples in $\mathcal{L}^t$, and $I(\cdot)$ is an indicator function. Moreover, t_y^t is a target value for a class y at time t, defined as

$$t_y^t = \frac{N_y^t+1}{N_y^t+2}, \tag{53}$$

where N_y^t is the number of labeled samples in $\mathcal{L}^t$ belonging to a class y at time t.

4.2 An expected error reduction function

One of the novelties in this chapter is the development of an expected error reduction function. It is defined as the amount of classification error reduced by an unlabeled sample x when the sample is included into the selection set $\mathcal{S}^t$, a set of unlabeled samples chosen for parameter update at time t. To evaluate such a quantity, we need to define an error measure first. While many candidates, such as Eq. (4), Eq. (5), and Eq. (6), exist as discussed in Section 2.1.1, our error measure is based on the confidence score defined in Eq. (47). In particular, suppose we have the true class label y_i for a sample x_i and a classification result $\delta(x_i;\boldsymbol{\theta}^t)$ with the corresponding confidence score $s_{conf}(\delta(x_i;\boldsymbol{\theta}^t))$. Then, the error measure $V:\mathcal{Y}\times\mathcal{Y}\rightarrow\mathbb{R}$ denoted as $V(\delta(x_i;\boldsymbol{\theta}^t),y_i)$ is given by

$$V(\delta(x_i;\boldsymbol{\theta}^t),y_i) = I(\delta(x_i;\boldsymbol{\theta}^t)=y_i)[1-s_{conf}(\delta(x_i;\boldsymbol{\theta}^t))] + I(\delta(x_i;\boldsymbol{\theta}^t)\neq y_i)s_{conf_{y_i}}(\delta(x_i;\boldsymbol{\theta}^t)), \tag{54}$$

where $I(\cdot)$ is an indicator function, and $s_{conf_{y_i}}(\delta(x_i;\boldsymbol{\theta}^t))$ is a variant of the confidence score that is given in Eq. (47), where the maximization term is replaced with the class conditional probability of a sample x_i for the true class label y_i, $P(x_i|y_i;\boldsymbol{\theta}^t)$. More precisely,

$s_{conf_{y_i}}(\delta(x_i;\boldsymbol{\theta}^t))$ is written as follows:

$$s_{conf_{y_i}}(\delta(x_i;\boldsymbol{\theta}^t)) = \frac{P(x_i|y;\boldsymbol{\theta}^t)}{P(x_i|y;\boldsymbol{\theta}^t) + P(x_i|y_i;\boldsymbol{\theta}^t)}|_{y=\delta(x_i;\boldsymbol{\theta}^t)}. \tag{55}$$

From Eq. (54), one can easily see that it is considered that there is no error if the output of a classifier for x_i,$\delta(x_i;\boldsymbol{\theta}^t)$, makes a correct decision with a high confidence score. On the other hand, the error measure will be the largest when the classifier produces an incorrect prediction result and assigns a zero value of class conditional probability to the ground-truth label y_i. One difficulty of using Eq. (54) to compute an error reduction directly is the fact that the ground-truth class label y_i for the sample x_i is unknown (x_i is an unlabeled sample). To tackle this issue, we compute an expected value of Eq. (54) over a class posterior probability of the sample x_i, $P(y|x,\boldsymbol{\theta}^t)$, as follows:

$$\begin{aligned}
\mathbb{E}_{y|x;\boldsymbol{\theta}^t} V(\delta(x_i;\boldsymbol{\theta}^t), y) &= \sum_{y=1}^{C} P(y|x_i;\boldsymbol{\theta}^t) V(\delta(x_i;\boldsymbol{\theta}^t), y) \\
&= P(y|x_i;\boldsymbol{\theta}^t)[1 - s_{conf}(\delta(x_i;\boldsymbol{\theta}^t))]|_{y=\delta(x_i;\boldsymbol{\theta}^t)} \\
&\quad + \sum_{y^- \in \mathcal{Y}\setminus y} P(y^-|x_i;\boldsymbol{\theta}^t) s_{conf_{y^-}}(\delta(x_i;\boldsymbol{\theta}^t))|_{y=\delta(x_i;\boldsymbol{\theta}^t)},
\end{aligned} \tag{56}$$

where the cardinality of a class label set $\mathcal{Y}$ is C. Given this definition of the expected error measure in Eq. (56), we can now define an expected error reduction for the sample x_i at time t, denoted as $s_{eer}(\delta(x_i;\boldsymbol{\theta}^t))$. By definition, $s_{eer}(\delta(x_i;\boldsymbol{\theta}^t))$ should measure how much the classification error of a current model is decreased when the sample x_i is included a selection set $\mathcal{S}^t$. Thus, we define it as the difference of the two expected error measures: one computed at time t with the parameter vector $\boldsymbol{\theta}^t$ and the other at time $t+1$ with $\boldsymbol{\theta}^{t+1}$, respectively. The mathematical formulation of $s_{eer}(\delta(x_i;\boldsymbol{\theta}^t))$ is thus given by

$$s_{eer}(\delta(x_i;\boldsymbol{\theta}^t)) = \mathbb{E}_{y|x;\boldsymbol{\theta}^t} V(\delta(x_i;\boldsymbol{\theta}^t), y) - \mathbb{E}_{y|x;\boldsymbol{\theta}^{t+1}} V(\delta(x_i;\boldsymbol{\theta}^{t+1}), y). \tag{57}$$

While Eq. (57) appears to be straightforward to compute (i.e., plug Eq. (56) into Eq. (57) and then simplify the resulting terms.), evaluating Eq. (57) is, in fact, computationally expensive because it depends upon a parameter vector at time t as well as that for time $t+1$. This requires us to perform training as many times as the total number of unlabeled samples in $\mathcal{U}^t$. To mitigate this difficulty, we make the following two assumptions:

- once we accept the prediction result given by a current model $\delta(x_i;\boldsymbol{\theta}^t)$ as the true class label for x_i, the confidence score corresponding to x_i at time $t+1$, $s_{conf}(\delta(x_i;\boldsymbol{\theta}^{t+1}))$, becomes unity.

- $P(y|x_i;\boldsymbol{\theta}^t) \approx P(y|x_i;\boldsymbol{\theta}^{t+1})$ for $\forall y \in \mathcal{Y}$.

One might, of course, argue the validity of the above assumptions. However, when the size of a labeled data set at time t, denoted as $\mathcal{L}^t$, is small while a classification model family is complex enough, the first assumption will be easily satisfied. The second assumption will also hold if the selection set $\mathcal{S}^t$ is small so that the labeled data set $\mathcal{L}^t$ remains relatively unchanged (i.e., $\mathcal{L}^t \approx \mathcal{L}^t \cup \mathcal{S}^t$); we set the maximum size of $\mathcal{S}^t$ to 3% of the number of labeled samples in $\mathcal{L}^t$ throughout the development of the proposed incremental learning framework. Given the above two assumptions, we can simplify Eq. (57) as follows:

$$\begin{aligned} s_{eer}(\delta(x_i;\boldsymbol{\theta}^t)) &= P(y|x_i;\boldsymbol{\theta}^t)[1 - s_{conf}(\delta(x_i;\boldsymbol{\theta}^t))]|_{y=\delta(x_i;\boldsymbol{\theta}^t)} \\ &\quad - \sum_{y^- \in \mathcal{Y}\setminus y} P(y^-|x_i;\boldsymbol{\theta}^t)[1 - s_{conf_{y^-}}(\delta(x_i;\boldsymbol{\theta}^t))]|_{y=\delta(x_i;\boldsymbol{\theta}^t)}, \end{aligned} \tag{58}$$

which is now a function of only the current parameter vector $\boldsymbol{\theta}^t$.

4.3 Robust estimation of the expected error reduction through an ensemble of classifiers

One thing to note for the expected error reduction defined in Eq. (58) is that the class posterior probability $P(y|x_i;\boldsymbol{\theta}^t)$ of a sample x_i used in Eq. (58) is not the true class posterior probability, which we denote it as $P^*(y|x_i)$, but an estimated value derived from a current model $\boldsymbol{\theta}^t$. In fact, the use of an estimate of the true posterior probability $P^*(y|x_i)$ might results in a biased expected error reduction. For example, suppose we have a binary-class classification problem, and $P(y|x_i;\boldsymbol{\theta}^t)$ is an unbiased estimator of $P^*(y|x_i)$. Suppose further that the labeled data set $\mathcal{L}^t$ is a set of randomly drawn samples given a marginal distribution of x, $P(x|\mu)$, where μ is its hyperparameter. Then, the expected value of $s_{eer}(\delta(x_i;\boldsymbol{\theta}^t))$ over the different sets of labeled data samples, $\mathcal{L}^t$s, can be written as follows:

$$\begin{aligned} \mathbb{E}_{\mathcal{L}^t}[s_{eer}(\delta(x_i;\boldsymbol{\theta}^t))] &= \mathbb{E}_{\mathcal{L}^t}[-2P^2(y|x_i;\boldsymbol{\theta}^t) + 3P(y|x_i;\boldsymbol{\theta}^t)]|_{y=\delta(x_i;\boldsymbol{\theta}^t)} - 1 \\ &= (-2P^{*2}(y|x_i) + 3P^*(y|x_i) - 1 - 2\sigma_P)|_{y=\delta(x_i;\boldsymbol{\theta}^t)}, \end{aligned} \tag{59}$$

where σ_P is the variance of $P(y|x_i;\boldsymbol{\theta}^t)$ with respect to $\mathcal{L}^t$s. From Eq. (59), it can be seen that $\mathbb{E}_{\mathcal{L}^t}[s_{eer}(\delta(x_i;\boldsymbol{\theta}^t))] < 0$ regardless of what value $P(y|x_i)|_{y=\delta(x_i;\boldsymbol{\theta}^t)}$ will be if σ_P is large. In other words, on average, the modeling accuracy will not be enhanced even after including the sample x_i into the learning process when the estimate of the posterior probability has a large variability. Note that in Eq. (59), we implicitly assume that $P(y|x_i;\boldsymbol{\theta}^t)$ is an unbiased estimate of $P^*(y|x_i)$. If $P(y|x_i;\boldsymbol{\theta}^t)$ is no longer an unbiased estimate of $P^*(y|x_i)$, Eq. (58) might reflect the actual contributions of unlabeled samples on reducing classification errors even more poorly.

To address this problem, we propose to use an ensemble of classifiers [48] with which we reduce the variance of an estimate of $P^*(y|x_i)$ by computing a sample mean of the posterior probability $P(y|x_i;\boldsymbol{\theta}^t)$ over classifiers in the ensemble. It is well-known that the variance of a classification result is inversely proportional to the number of classifiers used jointly to draw such an unified output[32]. Thus, given J distinct classifiers and the corresponding decision rules denoted as $\delta(x_i;\boldsymbol{\theta}^{(j)t})$ for $j = 1,\ldots J$, we compute a set of confidence scores, $s_{conf}(\delta(x_i;\boldsymbol{\theta}^{(j)t}))$ and a collection of estimates of $P^*(y|x_i)$ denoted as $P(y|x_i;\boldsymbol{\theta}^{(j)t})$ for individual classifiers for $j = 1,\ldots,J$. We then compute the expected error reduction for the k^{th} classifier $s_{eer}(\delta(x_i;\boldsymbol{\theta}^{(k)t}))$ as follows:

$$\begin{aligned} s_{eer}(\delta(x_i;\boldsymbol{\theta}^{(k)t})) &= P(y|x_i;\boldsymbol{\theta}^{(k^-)t})[1 - s_{conf}(\delta(x_i;\boldsymbol{\theta}^{(k)t}))]|_{y=\delta(x_i;\boldsymbol{\theta}^t)} \\ &\quad - \sum_{y^-\in\mathcal{Y}\setminus y} P(y^-|x_i;\boldsymbol{\theta}^{(k^-)t})[1 - s_{conf_{y^-}}(\delta(x_i;\boldsymbol{\theta}^{(k)t}))]|_{y=\delta(x_i;\boldsymbol{\theta}^t)}, \quad (60) \end{aligned}$$

where $P(y|x_i;\boldsymbol{\theta}^{(k^-)t})$ is an average of $P(y|x_i;\boldsymbol{\theta}^{(j)t})$ over $j = 1,\ldots,k-1,k+1,\ldots,J$ (i.e., $P(y|x_i;\boldsymbol{\theta}^{(k^-)t})$ is an average of the posterior probabilities of all classifiers in an ensemble except for the k^{th} one.) with which we aim at reducing the correlation between the value of posterior probability and that of confidence score when computing the expected error reduction. In fact, it can be empirically shown that the large amount of the correlation between the posterior probability and the confidence score creates a bias for the resulting expected error reduction value, which in turn, creating a selection set $\mathcal{S}^t$ that does not help to improve classification accuracy. Now, given Eq. (60), our expected error reduction

function is defined as an average of the individual expected error reductions as follows:

$$s_{eer}(\delta(x_i;\boldsymbol{\theta}^t)) = \frac{1}{J}\sum_{j=1}^{J} s_{eer}(\delta(x_i;\boldsymbol{\theta}^{(j)t})). \tag{61}$$

To understand the behaviors of Eq. (61), we plot the values of the expected error reduction function given by Eq. (61) in Figure 10 against different values of class posterior probabilities, where the size of the ensemble and the number of classes are assumed to be both two (i.e., $J = 2$ and $C = 2$). Note that when $C = 2$, the confidence score $s_{conf}(\delta(x_i;\boldsymbol{\theta}^t))$ is the same as the class posterior probability $P(y|x_i;\boldsymbol{\theta}^t)$ evaluated at $y = \delta(x_i;\boldsymbol{\theta}^t)$, as discussed in Section 4.1. Therefore, in this example, the decision rule $\delta(x_i;\boldsymbol{\theta}^{(j)t})$ for a sample x_i and the j^{th} classifier is given by

$$\delta(x_i;\boldsymbol{\theta}^{(j)t}) = \{y|P(y|x_i,\boldsymbol{\theta}^{(j)t}) \geq 0.5,\ y = 1,2\} \tag{62}$$

Now, in Figure 10, note the upper-right and -left corners where the expected error reduction attains its maximum value. These points satisfy the following properties: (a) the classification results for both classifiers are the same (i.e., $\delta(x_i;\boldsymbol{\theta}^{(1)t}) = \delta(x_i;\boldsymbol{\theta}^{(2)t})$), and (b) the class posterior probabilities for individual classifiers, denoted as $P(y|x_i;\boldsymbol{\theta}^{(1)t})$ and $P(y|x_i;\boldsymbol{\theta}^{(2)t})$ respectively, are maximally disagreeing with each other subject to the constraint that $\delta(x_i;\boldsymbol{\theta}^{(1)t}) = \delta(x_i;\boldsymbol{\theta}^{(2)t})$. More precisely, the following holds:$P(y|x_i;\boldsymbol{\theta}^{(1)t}) = 0.5$ and $P(y|x_i;\boldsymbol{\theta}^{(2)t}) = 1$, or vice versa. Interestingly enough, because of the second property, our expected error reduction function can be considered as an adapted version of the *query-by-committee* (QBC) sampling strategy proposed for *active learning* (see Section 2.3.1) to semi-supervised cases. Recall that in the QBC sampling strategy, a sample that the committee members differ from each other in their prediction results the most should be selected because such a sample will maximally improve the overall modeling accuracy. The difference between the QBC sampling strategy and our expected error reduction is the fact that the latter requires the prediction outcomes to be matched among classifiers while the former does not.

The analogy between the proposed expected error reduction and the QBC sampling strategy is still valid even when there are more than two classifiers in an ensemble (i.e.,

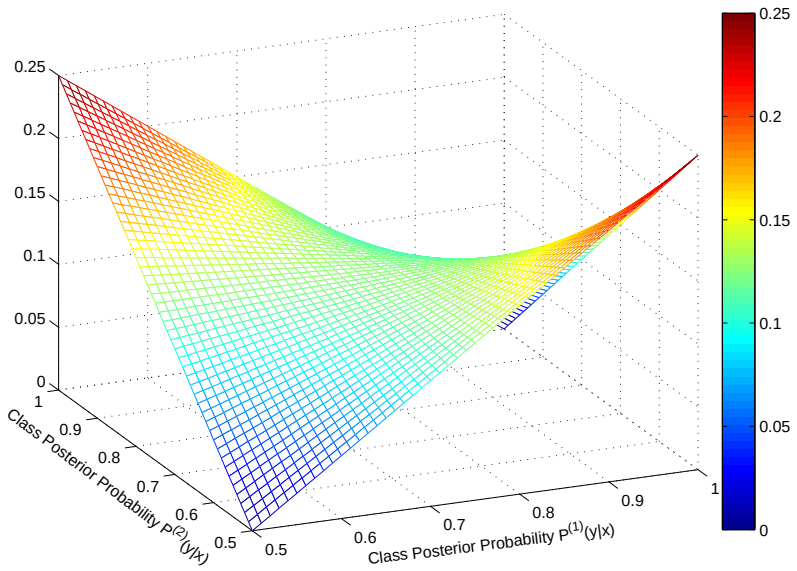


Figure 10: The values of the expected error reduction defined in Eq. (61) when the size of an ensemble is two. We also assume that the number of classes is also two. Note the upper-right and -left corners where the maximum values of the expected error reduction are attained. Those points are where two class posterior probabilities are maximally disagreeing with each other.

$J > 2$). To see this, let us focus on the fact that in Figure 10 where the size of an ensemble is two, both the minimum and maximum values of the expected error reduction are attained at the extreme points of the bended surface (i.e., either $P(y|x_i;\boldsymbol{\theta}^{(j)t}) = 1$ or $P(y|x_i;\boldsymbol{\theta}^{(j)t}) = 0.5$ for $j = 1, 2.$). In fact, one can show that this characteristic still holds as long as we have a finite number of classifiers in an ensemble. Therefore, one can identify in which conditions, the minimum and maximum values of the expected error reduction are attained by enumerating all possible combinations of such extreme points for $j = 1, \ldots, J$. We summarize the findings as follows:

$$s_{eer}(\delta(x^*;\boldsymbol{\theta}^t)) \geq s_{eer}(\delta(x_i;\boldsymbol{\theta}^t)), \tag{63}$$

for all samples $x_i \in \mathcal{D}$ if

$$(P(y|x^*;\boldsymbol{\theta}^{(1)t}), \ldots, P(y|x^*;\boldsymbol{\theta}^{(J)t})) = (\underbrace{1, \ldots, 1}_{\frac{J}{2}}, \underbrace{0.5, \ldots, 0.5}_{\frac{J}{2}}) \tag{64}$$

when J is even, or if

$$\begin{aligned}(P(y|x^*;\boldsymbol{\theta}^{(1)t}),\ldots,P(y|x^*;\boldsymbol{\theta}^{(J)t})) &= (\underbrace{1,\ldots,1}_{\frac{J-1}{2}},\underbrace{0.5,\ldots,0.5}_{\frac{J-1}{2}+1}) \\ &\text{or} \\ (P(y|x^*;\boldsymbol{\theta}^{(1)t}),\ldots,P(y|x^*;\boldsymbol{\theta}^{(J)t})) &= (\underbrace{1,\ldots,1}_{\frac{J-1}{2}+1},\underbrace{0.5,\ldots,0.5}_{\frac{J-1}{2}})\end{aligned} \tag{65}$$

when J is odd, where $x^* \in \mathcal{D}$ is a maximizer of the expected error reduction. The conditions presented in Eqs. (64) and (65) clearly demonstrate the similarity of the expected error reduction to the QBC sampling strategy with the following remarks: (a) the maximum expected error reduction is achieved if a half of J classifiers are completely sure on their classification results, while the remaining half of the classifiers are maximally inconfident in their decisions, and (b) to achieve the maximum expected error reduction, the classification results of individual models in an ensemble should be consistent. Note that the order of classifiers in Eqs. (64) and (65) does not matter for obtaining a maximum value, although we arrange them in an ascending order for an easier presentation. Note further that Eq. (63) is true as long as J is finite. When $J \to \infty$, one can also prove that $\max_{x\in\mathcal{D}} s_{eer}(\delta(x;\boldsymbol{\theta}^t))$ approaches to $\frac{1}{8}$ and the maximizer x^* is the sample that the corresponding posterior probabilities $P(y|x^*;\boldsymbol{\theta}^{(j)t})$ for $j = 1,\ldots,\infty$ are all equal to 0.75.

4.4 The proposed algorithm

So far, we have discussed how to evaluate (a) the confidence score and (b) the expected error reduction. Given these two quantities, in this section, we present our proposed semi-supervised incremental learning framework with which a selection set $\mathcal{S}^t$ is sampled from an unlabeled data set $\mathcal{U}^t$ in a way that the maximal reduction of the modeling error for a current classifier is achieved. To this end, given a sample x_i, we can first treat the expected error reduction of at a certain time t, $s_{eer}(\delta(x_i;\boldsymbol{\theta}^t))$, as a penalty term of the confidence score of the sample, $s_{conf}(\delta(x_i;\boldsymbol{\theta}^t))$ similar to the preliminary work presented in [17]. In particular, we compute a weighted score between $s_{eer}(\delta(x_i;\boldsymbol{\theta}^t))$ and $s_{conf}(\delta(x_i;\boldsymbol{\theta}^t))$, resulting

in a *selection score* defined as

$$s(\delta(x_i;\boldsymbol{\theta}^t)) = (1-\gamma)s_{eer}(\delta(x_i;\boldsymbol{\theta}^t)) + \gamma s_{conf}(\delta(x_i;\boldsymbol{\theta}^t)), \tag{66}$$

where γ is a convex combination coefficient, to create the selection set $\mathcal{S}^t$. Note that we redefine the confidence score $s_{conf}(\delta(x_i;\boldsymbol{\theta}^t))$ in Eq. (66) to take advantage of an ensemble of classifiers as

$$s_{conf}(\delta(x_i;\boldsymbol{\theta}^t)) = \frac{1}{J}\sum_{j=1}^{J} s_{conf}(\delta(x_i;\boldsymbol{\theta}^{(j)t})), \tag{67}$$

where $s_{conf}(\delta(x_i;\boldsymbol{\theta}^{(j)t}))$ represents the confidence score for the j^{th} classifier defined as the one in Eq. (47), and J is the total number of classifiers in the ensemble.

We can understand the behaviors of Eq. (66) by plotting the values of $s(\delta(x_i;\boldsymbol{\theta}^t))$ for the various values of class posterior probabilities in Figure 11 similar to what is shown in Figure 10. Here, we set the convex combination coefficient γ to 0.5. To contrast Figure 11

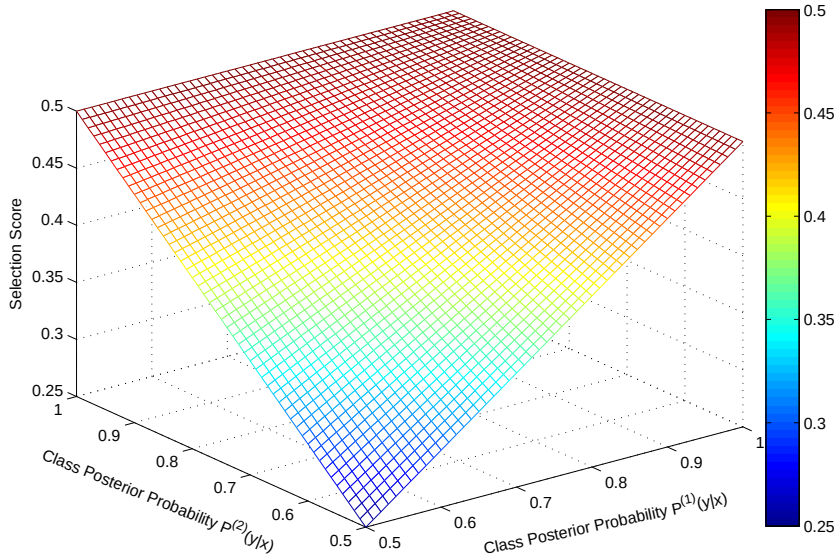


Figure 11: The values of the selection scores $s(\delta(x_i;\boldsymbol{\theta}^t))$ given by Eq. (66) when the size of an ensemble and the number of classes are both assumed to be two and $\gamma = 0.5$. Note that the maximum value is attained not only when $P^{(1)}(y|x)$ and $P^{(2)}(y|x)$ disagree the most (the upper-right and -left corners), but also when $P^{(1)}(y|x)$ and $P^{(2)}(y|x)$ are all equal to unity (the upper-middle corner).

with Figure 10, we assume that the size of the ensemble and the number of classes are all set to two as in Figure 10. Now, it is shown that, in Figure 11, the selection score $s(\delta(x_i;\boldsymbol{\theta}^t))$

attains its maximum value not only when the class posterior probabilities $P(y|x_i;\boldsymbol{\theta}^{(1)t})$ and $P(y|x_i;\boldsymbol{\theta}^{(2)t})$ are disagreeing the most as in the case with Figure 10, but also when both of the classifiers are almost certain in their decisions. This implies that the corresponding selection set $\mathcal{S}^t$ constructed based on the weighted score will choose samples with a high confidence score from *at least* one classification model. On the other hand, if $\mathcal{S}^t$ is collected based on Figure 10 (i.e., according to the expected error reduction defined in Eq. (61) only), the samples in $\mathcal{S}^t$ will have a high confidence score *only* from a single classifier. This observation implies the following: the convex combination parameter γ actually determines the number of classifiers that have to be certain in their predictions for an unlabeled sample x_i to be included into $\mathcal{S}^t$.

To see this, in Figure 12, a similar mesh-grid plot of the selection score $s(\delta(x_i;\boldsymbol{\theta}^t))$ to Figure 11 is drawn when the convex combination parameter γ is now set to 0.42. Except for the value of the convex combination parameter, other configurations are exactly the same as those in Figure 11. Confirming our argument, one can easily see from Figure 12 that when only one classifier has a high confidence score (and the other is the least certain for its decision), unlabeled samples will be most likely to be selected for $\mathcal{S}^t$. Now, one might as well want to collect unlabeled samples whose selection scores are larger than a certain threshold τ, instead of looking at only those extreme cases. Thus, we additionally draw lines in Figure 12 to indicate the areas where the combined scores are greater than 93.5% of a maximum value. The strategy to construct a selection set $\mathcal{S}^t$ here is then as follows: (a) set the threshold value as 93.5% of the maximum value (e.g., 0.43 in Figure 12), and (b) from a set of available unlabeled data samples at time t, $\mathcal{U}^t$, collect samples in which the corresponding selection scores are greater than the threshold, and include them into $\mathcal{S}^t$. Given this strategy, from the Figure 12, one can easily see that $\mathcal{S}^t$ will now include samples that the corresponding posterior probabilities $P(y|x_i;\boldsymbol{\theta}^{(1)t})$ and $P(y|x_i;\boldsymbol{\theta}^{(2)t})$ are equal to the following values: (a) $P(y|x_i;\boldsymbol{\theta}^{(1)t}) = 0.94$ and $P(y|x_i;\boldsymbol{\theta}^{(2)t}) = 0.5$, (b) $P(y|x_i;\boldsymbol{\theta}^{(1)t}) = 1$ and $P(y|x_i;\boldsymbol{\theta}^{(2)t}) = 0.87$, etc.

When there are more than two classifiers in an ensemble, one can still build a similar argument to that with two classifiers, although a more subtle analysis is required. To this end,

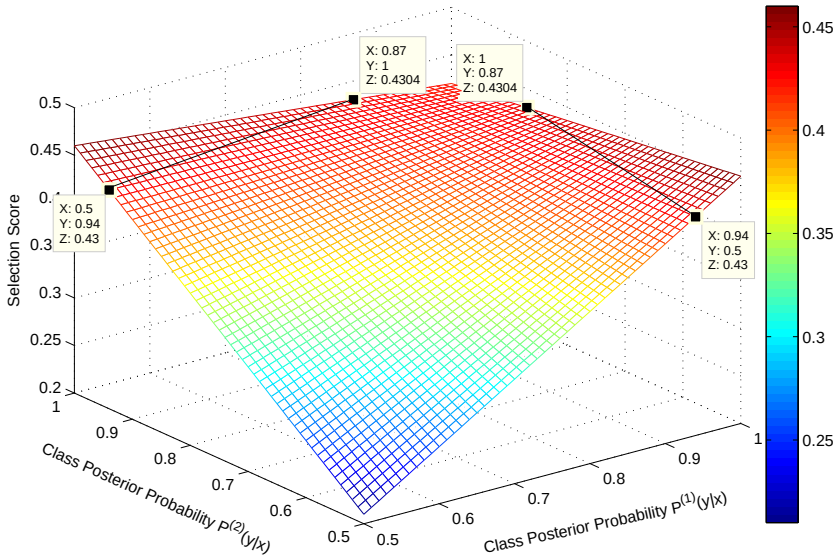


Figure 12: The values the selection scores $s(\delta(x;\boldsymbol{\theta}^t))$ when the size of an ensemble and the number of classes are both assumed to be two and $\gamma = 0.42$. Lines indicate the boundaries where $s(\delta(x;\boldsymbol{\theta}^t))$ is 93.5% of its maximum value, where the maximum of $s(\delta(x;\boldsymbol{\theta}^t))$ is 0.46. Thus, $\tau = 0.935$, we will select unlabeled samples such that the corresponding selection score $s(\delta(x;\boldsymbol{\theta}^t)) \geq 0.43$

let us first recall that Eq. (63) says that the maximum expected error reduction is achieved when the prediction results from a half of the classifiers in an ensemble are absolutely sure (i.e., $P(y|x_i;\boldsymbol{\theta}^{(j)t}) = 1$ for $1 \leq j \leq \frac{J}{2}$ if J is an even number.), while the remaining half of the classifiers are maximally uncertain on their decisions (i.e., $P(y|x_i;\boldsymbol{\theta}^{(j)t}) = 0.5$ for $\frac{J}{2} + 1 \leq j \leq J$ when J is even.). Now, let us consider a more general case in which at least one classifier in the ensemble has a class posterior probability at those extreme points (i.e. $P(y|x_i;\boldsymbol{\theta}^{(j)t}) = 0.5$ or $P(y|x_i;\boldsymbol{\theta}^{(j)t}) = 1$). Then, it can be easily seen that the expected error reduction becomes piecewise linear in such cases. Moreover, given the same condition, the confidence score given by Eq. (67) also becomes piecewise linear, which makes the weighted score $s(\delta(x_i;\boldsymbol{\theta}^t))$ piecewise linear as well. Note that a convex combination of two piecewise linear functions is also piecewise linear. Whence, it is sufficient to enumerate all the possible combinations of the extreme values of class posterior probabilities for individual classifiers in the ensemble, similar to what we did for Eq. (63), to unfold the condition when the

maximum value of $s(\delta(x_i;\boldsymbol{\theta}^t))$ is obtained for different values of convex combination parameters γ. Figure 13 summarizes such results where we plot the number of classifiers that the associated class posterior probabilities equal to one to make the selection score $s(\delta(x_i;\boldsymbol{\theta}^t))$ the maximum as a function of the weighting coefficient γ. We also change the total number of classifiers in an ensemble, J, among 2, 4, 8, and 16. In essence, Figure 13 demonstrates

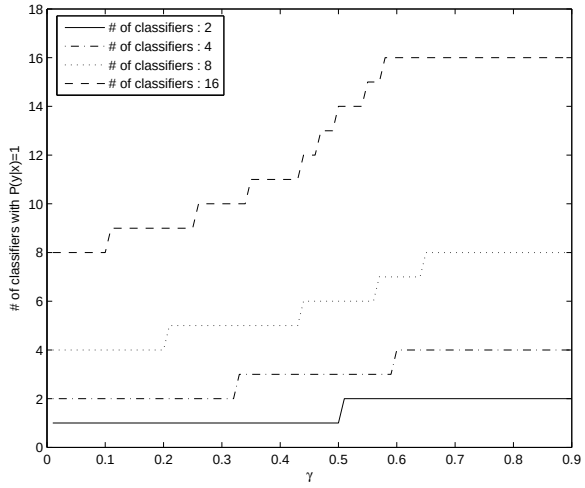


Figure 13: The number of classifiers with $P(y|x;\boldsymbol{\theta}^t) = 1$ to attain the maximum value of the selection score $s(\delta(\cdot;\boldsymbol{\theta}^t))$ against various values of γ while the number of total classifiers is varied from 2 to 16.

how to control the number of classifiers that need to produce confident predictions for an unlabeled sample x_i to be chosen for S^t. In particular, when there are eight classifiers in an ensemble and γ is set to 0.5, we prefer to incorporate samples into $\mathcal{S}^t$ if six out of the eight classifiers are fairly comfortable in their classification decisions for those samples. On the other hand, when the size of an ensemble is equal to four, it is most likely to choose samples for $\mathcal{S}^t$ when all of four classifiers generate high confidence scores for them if γ is set to 0.7. In sum, Figure 13 provides very useful information to choose a parameter γ based on the amount of assurance that a designer wants to pursue before setting off an incremental learning process. One interesting thing to note in Figure 13 is that no matter what value γ is assigned to, at least a half of the classifiers in an ensemble have to be sure on their predictions. This is intuitively pleasing because conceptually, it does not make sense to exploit such a sample for training if more than a half of the classifiers do not have confident

class label predictions for the sample.

So far, we have introduced two parameters, a threshold τ and a convex combination parameter γ, and discussed the characteristics of the selection score in different parameter settings. One potential problem of creating a selection set $\mathcal{S}^t$ with the selection score and the parameters τ and γ is that it does not guarantee that the number of samples that belongs to each class in such a selection $\mathcal{S}^t$ is proportional to a class prior distribution $P(y)$ for $y \in \mathcal{Y}$. While this is not necessary to have the convergence property of incremental learning (i.e., the performance of an initial model is enhanced through the learning procedures.), the selection set $\mathcal{S}^t$ tends to contain too many samples for a major class, which might cause the *class imbalance problem* and thus, resulting in inferior performance at the end. To tackle this issue, in the proposed framework, we first compute a maximum likelihood estimate of the *class prior distribution* $P(y)$, denoted as $\hat{P}(y;\boldsymbol{\theta}^t)$, as follows:

$$\hat{P}(y;\boldsymbol{\theta}^t) = \frac{1}{J}\frac{1}{|\mathcal{D}|}\sum_{j=1}^{J}\sum_{x_i \in \mathcal{D}} P(y|x_i;\boldsymbol{\theta}^{(j)t}), \tag{68}$$

where $|\mathcal{D}|$ is the cardinality of a training data set $\mathcal{D}$ and J is the number of classifiers in an ensemble. Given Eq. (68), the procedures are then as follows: (a) we randomly draw a candidate class y according to Eq. (68), (b) among the samples predicted as being in the class y, an unlabeled sample $x_{u_i} \in \mathcal{U}^t$ is picked in a descending order of the corresponding selection score $s(\delta(x_{u_i};\boldsymbol{\theta}^t))$, where $\mathcal{U}^t$ corresponds to a set of available unlabeled samples at time t, and the subscript u_i is used to represent the i^{th} unlabeled sample in $\mathcal{U}^t$. Readers should differentiate it from the plain subscript i, which denotes the i^{th} sample in the entire training data set $\mathcal{D}$.

In addition to the use of an estimate of class prior distribution, $\hat{P}(y;\boldsymbol{\theta}^t)$, we also make a further effort to create a more robust selection set $\mathcal{S}^t$ with using a Zipf distribution. A Zipf distribution is an empirical distribution with a discrete random variable $X > 0$ in which the probability of X being x is inversely proportional to the value of x, where the random variable X typically represents a ranking of a certain entity. In the proposed framework, therefore, we first evaluate the ranking of the selection scores among those unlabeled samples that are predicted to be in a class y given $\mathcal{U}^t$. We denote it as $rank_y\{s(\delta(\cdot;\boldsymbol{\theta}^t))\}$. Then

given a candidate class y chosen based on Eq. (68), we select a sample $x_{u_i} \in \mathcal{U}^t$ according to the probability distribution given by

$$P_{zipf}(x_{u_i}; y) = \frac{1}{Z} \cdot \frac{1}{rank_y\{s(\delta(x_{u_i}; \boldsymbol{\theta}^t))\}^\eta} I(y = \delta(x_{u_i}; \boldsymbol{\theta}^t)), \tag{69}$$

where $I(\cdot)$ is an indicator function, Z represents a normalization parameter, and $\eta \geq 0$ is a parameter that determines the characteristics of the probability distribution in Eq. (69). In particular, when η is large, only the top ranked samples will be picked, while $\eta = 0$, all samples whose corresponding selection scores are above the threshold τ (i.e., $s(\delta(x_i; \boldsymbol{\theta}^t)) \geq \tau$) will be selected uniformly.

In sum, we conclude this section by describing the detailed algorithmic procedures of the proposed semi-supervised learning framework in Algorithm 3.

4.5 Experimental results

To evaluate the proposed technique, we prepared two data sets used in Chapter 3; the USPS data set for handwritten digit recognition, and the COIL-100 data set for object recognition. Features were extracted in the same way as discussed in Chapter 3. Out of the total 1500 samples, we randomly chose 300 samples for testing and use the remaining 1200 samples for training. Among 1200 training samples, we further selected 5%, 10%, and 20% of them and made the initially labeled data sets, while the remaining sets of samples were treated as unlabeled data sets.

There were mainly four parameters to set: (a) the desired size of a selection set $\mathcal{S}^t$, denoted as k_u^t, (b) the threshold τ, (c) a positive constant η for the Zipf distribution, (d) and the convex combination parameter γ. For k_u^t, it was set to three percent of the size of the labeled data set at time t, $\mathcal{L}^t$, throughout all experiments. Because it is somewhat unrealistic to have a validation set for parameter tuning, the other parameters were determined based on the performance improvement on test sets for a first few iterations, say 5-10, depending on the size of an initially labeled data set (i.e., less iterations for a larger initial label set).

As for the classifiers used in an ensemble, we trained kernelized MFoM classifiers presented in Chapter 3 and a regularized spectral clustering technique based nearest neighbor

Algorithm 3 The proposed semi-supervised incremental learning algorithm

prepare $\mathcal{U}^0$ and $\mathcal{L}^0$
initialize $\boldsymbol{\theta}^0$ with $\mathcal{L}^0$
$t \leftarrow 0$
$C \leftarrow$ the number of classes
repeat
  compute Eq. (67), Eq. (61) and Eq. (66) for all samples in $\mathcal{U}^t$
  estimate the class prior distribution using all samples in $\mathcal{U}^t$ based on Eq. (68)
  $k_u^t \leftarrow$ the number of samples to be selected at time t
  $\mathcal{U}_y^t \leftarrow \{x \,|x \in \mathcal{U}^t,\ \delta(x;\boldsymbol{\theta}^t) = y,\ s(\delta(x;\boldsymbol{\theta}^t)) \geq \tau\}$ for $y = 1, \ldots, C$
  $N_{u_y^t} \leftarrow |\mathcal{U}_y^t|$ for $y = 1, \ldots, C$
  $\tilde{\mathcal{U}}_y^t \leftarrow \{x_{(i)} \,|x_{(i)} \in \mathcal{U}_y^t,\ s(\delta(x_{(1)}, \boldsymbol{\theta}^t)) \geq s(\delta(x_{(2)}, \boldsymbol{\theta}^t)) \geq \cdots \geq s(\delta(x_{(N_{u_y^t})}, \boldsymbol{\theta}^t))\}$
  compute a Zipf distribution based on Eq. (69) for all $\tilde{\mathcal{U}}_y^t, y = 1, \ldots, C$
  while $|\mathcal{S}^t| < k_u^t$ **do**
    pick a candidate class m according to the estimated class prior distribution
    if $|\tilde{\mathcal{U}}_m^t| \neq 0$ **then**
      pick x according to the computed Zipf distribution for the class m
      $\tilde{\mathcal{U}}_m^t \leftarrow \tilde{\mathcal{U}}_m^t \backslash x$
    else
      continue
    end if
  end while
  $\mathcal{L}^{t+1} \leftarrow \mathcal{L}^t \cup \mathcal{S}^t$
  $\mathcal{U}^{t+1} \leftarrow \mathcal{U}^t \backslash \mathcal{S}^t$
  update $\boldsymbol{\theta}^{t+1}$ with $\mathcal{L}^{t+1}$
  if $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t$ **then**
    break
  else
    $t \leftarrow t + 1$
  end if
until $|\mathcal{U}^t| = 0$

(NN) classifier originally presented in [114]. The use of such a regularized spectral clustering technique can also be found in [17]. The parameters used to train the kernelized MFoM classifiers were simply taken from those used in Chapter 3 to minimize the efforts required for parameter adjustment. For the regularized spectral clustering based NN classifier, there were mainly two parameters to tune: (a) the size of nearest neighbors and (b) a weight parameter to adjust the balance between the initial labels and the steady-state labels. We set these parameters to 10 and 0.8, respectively, throughout all experiments for simplicity. For more information regarding what these parameters are, please refer to [114]. Since the kernelized MFoM classifier did not produce class posterior probabilities, we estimated them

from the class score function $g_y(x_i)$ for $y = 1, \ldots, C$ (please refer to Chapter 3 for the definition of the class score function.) by assuming that the class conditional probability $P(x_{u_i}|y; \boldsymbol{\theta}^t)$ for a sample $x_{u_i} \in \mathcal{U}^t$ could be written as a log-linear of $g_y(x_i)$ as follows:

$$P(x_{u_i}|y; \boldsymbol{\theta}^t) = \frac{1}{Z(a)} e^{a g_y(x_{u_i}) + b_y}, \tag{70}$$

where a and b_y were parameters to be estimated, where the subscript y represents the dependency of the offset parameter b on a class label y, and $Z(a)$ was a normalization term, which was a function of a. The parameters a and b_y were found based on a maximum likelihood (ML) criterion. In particular, we first defined the class posterior probability for a sample x, $P(y|x; \boldsymbol{\theta}^t)$ using an uninformative prior and Bayes formula as

$$P(y|x_i; \boldsymbol{\theta}^t) = \frac{P(x_i|y; \boldsymbol{\theta}^t)}{\sum_{y'=1}^{C} P(x_i|y'; \boldsymbol{\theta}^t)}. \tag{71}$$

Then, we formulated the objective function to find a and b_y for $y = 1, \ldots, C$ given by

$$\max_{a, b_1, \ldots, b_C \in \mathbb{R}} \sum_{y=1}^{C} \sum_{i=1}^{N_l^t} t_y^t \log P(y|x_{l_i}; \boldsymbol{\theta}^t) I(y_{l_i} = y) \tag{72}$$

where N_l^t was the number of labeled data samples in $\mathcal{L}^t$, a set of available labeled samples at time t, and $I(\cdot)$ was an indicator function. Moreover, t_y^t was a target value for a class y at time t, defined as

$$t_y^t = \frac{N_y^t + 1}{N_y^t + 2}, \tag{73}$$

where N_y^t was the number of samples that belong to a class y in $\mathcal{L}^t$.

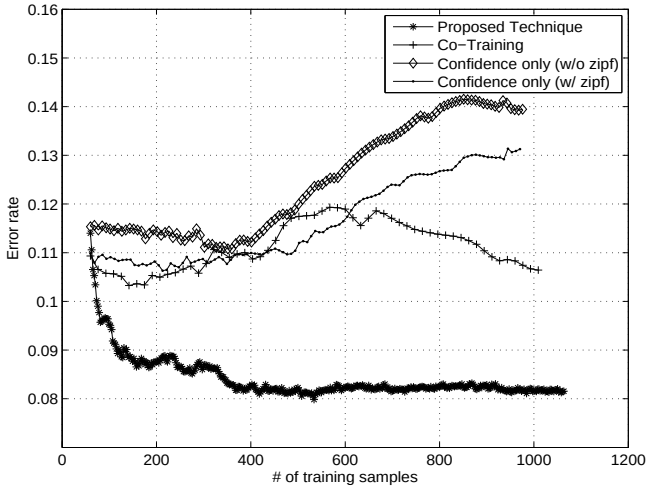
4.5.1 Comparisons with other semi-supervised incremental learning algorithms

In this section, we demonstrate the effectiveness of our proposed technique by making comparisons with two baseline systems, a confidence score based method and a Co-training method. To be able to make comparisons with the Co-training method, we have set the number of classifiers in an ensemble to two for the proposed framework as well. Moreover, to ensure the Co-training method produces its best results, the classifiers in the ensemble were chosen to be different, namely a kernelized MFoM classifier and a spectral clustering technique based NN classifier. Note that in Section 4.5.2, we experimented with an ensemble

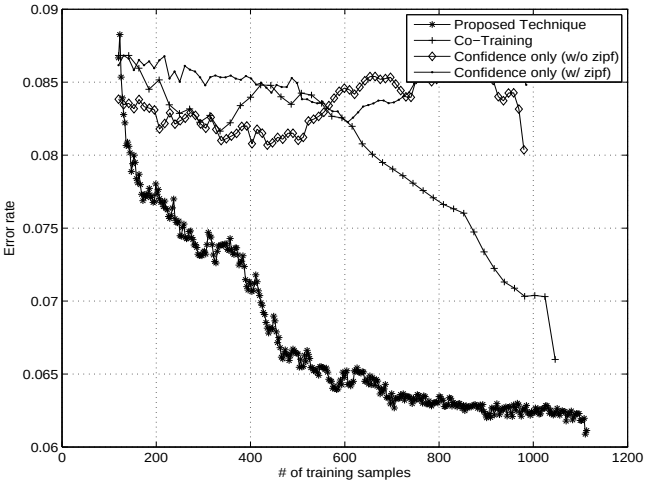
consisting of only kernel MFoM classifiers. For fair comparisons, class prior distributions and Zipf distributions were exploited in both baseline systems as well. Moreover, the parameters associated with the baseline systems were determined in the same way as the one used for the proposed technique (i.e., we chose parameters based on the performance improvement on a test set after a first few iterations). While varying the size of initially labeled data set from 5%, 10%, and 20% of the entire training data to see the effects of different sizes of initial sets, we illustrate the performance comparison curves on the USPS data set in Figure 14.

In Figure 14, it is clearly seen that the proposed technique outperforms all of the baseline systems by a large margin. Specifically, comparing the bottom curves (i.e, the proposed technique) in Figures 14-(a), -(b), and -(c), with the two curves in the top (i.e., the confidence score based methods), the proposed technique achieves more than 25% relative error rate reduction in all cases. Comparing the performances among different sizes of initial sets, when the size of an initial labeled set is 5% of the training data set, the proposed technique is able to show 26% of relative error rate reduction compared to its initial performance. Even when the size of an initial labeled set is increased to 20%, the proposed technique still shows 25.8% of relative error rate reduction over the initially trained system. On the other hand, when the size of an initial labeled set is small (i.e. 5%), both of the baseline systems cannot show any performance improvement. At first glance, it is somewhat surprising to see the Co-training does not perform well. However, this can be understood by the fact that two models (i.e. the kernelized MFoM learning approach and the regularized spectral clustering technique based NN classifier) have been trained on the same feature. Recall that for the Co-training to succeed, two classifiers should be sufficiently different initially. It is also interesting to see the rate of performance improvement is much higher for the proposed technique than the baseline systems. This is mainly due to the fact that the proposed technique actively searches for unlabeled samples with high values of expected error reduction, while the baseline systems simply wait for those samples to be picked up.

Similarly, Figure 15 illustrates performance comparison curves on the COIL-100 data set. Again, in Figure 15, the proposed technique outperforms all the baseline systems by a

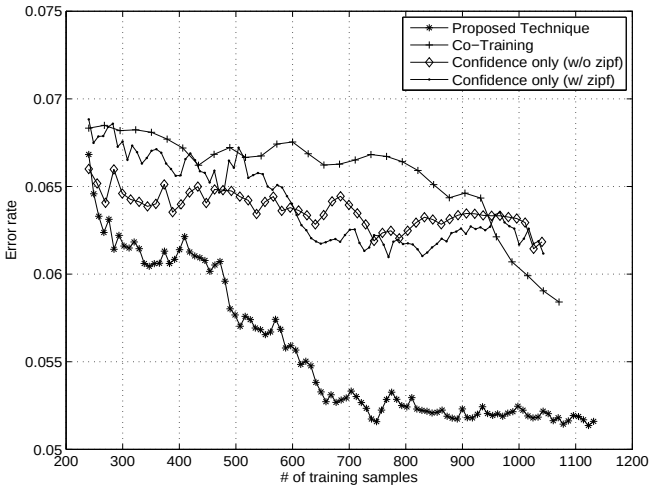


(a) Performance comparison curves when the size of an initially labeled set is 5 percent of the entire training data. The bottom curve is for the proposed framework and those upper two curves are for the confidence score only methods with or without the use of a Zipf distribution. The middle curve is for the Co-training method. As seen here, the proposed technique outperforms the baseline systems by a large margin.



(b) Performance comparison curves when the size of an initially labeled set is 10 percent of the entire training data. The bottom curve is for the proposed framework and those upper two curves are for the confidence score only methods with or without the use of a Zipf distribution. The middle curve is for the Co-training method. Still the proposed framework outperforms the baseline systems while the gap between the proposed technique and the Co-training is now narrowed a little bit.

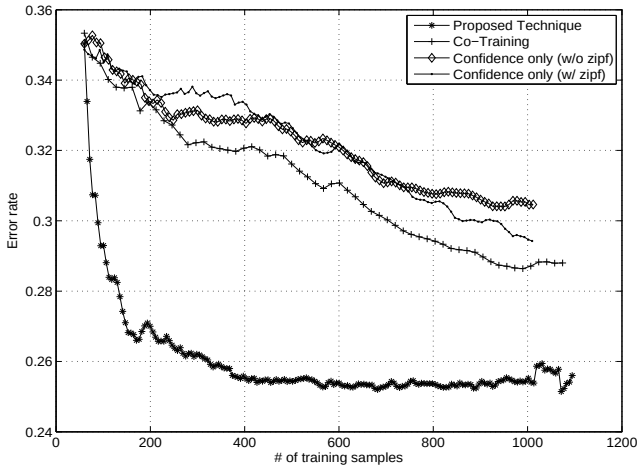
Figure 14
continued..



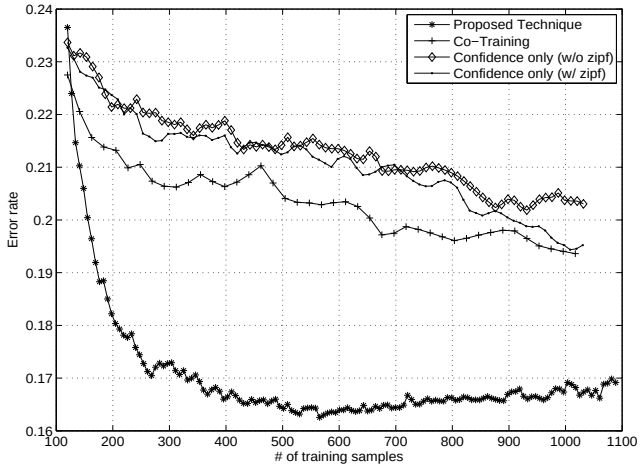
(c) Performance comparison curves when the size of an initially labeled set is 20 percent of the entire training data. The bottom curve is for the proposed framework and those two curves in the middle are for the confidence score only methods with or without the use of a Zipf distribution. The upper curve is for the Co-training method. The proposed framework outperforms the baseline systems with 20 percent of the labeled data set, while all the baseline systems are able to improve their intial models.

Figure 14 continued: Performance comparison curves between the baseline systems and the proposed technique for the USPS data set. x-axis represents the number of training samples that have been incoporated so far and y-axis represents an error rate. The sizes of initial labeled data sets for above features are as follows: (a) 5 percent, (b) 10 percent, and (c) 20 percent of the entire training data.

large margin (see the two curves in the top and the bottom curve). Unlike the case with the USPS data set, for the COIL-100 data set, the Co-training is able to show its effectiveness, but the amount of improvement that the Co-training shows is small compared to that of the proposed technique. The Co-training becomes somewhat comparable to the proposed technique when the size of the initial label set is large enough (i.e., 20%). However, first, the proposed framework still outperforms the Co-training methods, and second, having a large set of initially labeled samples (i.e., 20% of the entire training data in this case) is not always achievable. Compared to the performances of initially given models, the proposed technique show 28.1% and 31.7% of relative error rate reduction when the sizes of initial labeled sets are 5% and 20%, respectively. In this data set, the rate of performance improvement for the proposed technique is again significantly higher than that of the baseline systems.

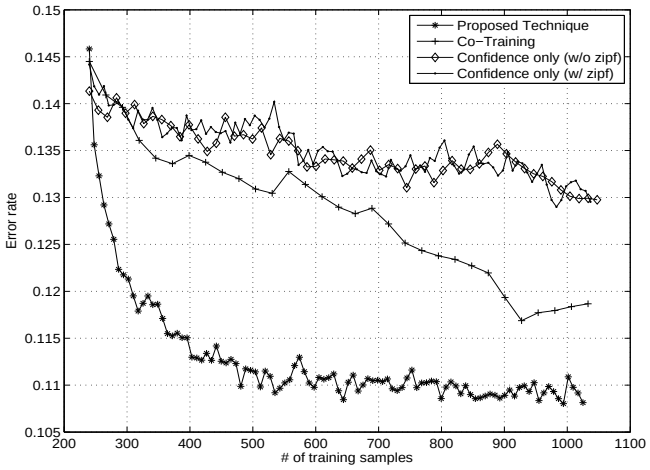


(a) Performance comparison curves when the size of an initially labeled set is 5 percent of the entire training data. The bottom curve is for the proposed framework and those upper two curves are for the confidence score only methods with or without the use of a Zipf distribution, respectively. The middle curve is for the Co-training method. As seen here, while the Co-training method shows some improvement over the confidence score based method, the proposed technique outperforms both of the baseline systems by a large margin.



(b) Performance comparison curves when the size of an initially labeled set is 10 percent of the entire training data. The bottom curve is for the proposed framework and those upper two curves are for the confidence score only methods with or without the use of a Zipf distribution, respectively. The middle curve is for the Co-training method. Similar patterns to the above figure can be found here; the proposed technique outperforms the baseline systems by a large margin.

Figure 15
continued..



(c) Performance comparison curves when the size of an initially labeled set is 20 percent of the entire training data. The bottom curve is for the proposed framework and those upper two curves are for the confidence score only methods with or without the use of a Zipf distribution, respectively. The middle curve is for the Co-training method. As it can be seen, the gap of the performance curves between the Co-training method and the proposed technique is narrowed. Still, the proposed technique outperforms the baseline systems.

Figure 15 continued: Performance comparison graphs between the baseline systems and the proposed framework for the COIL-100 data set. We tested with three different sizes of initially labeled data sets; 5, 10, 20 percent of the entire training data set, each of which corresponds to (a), (b) and (c), respectively. In all cases, the proposed technique outperforms the baseline systems with a large margin, while as the size of initial label set grows, the gap between the performances of the proposed technique and the baseline systems is narrowed. x-axis represents the number of training samples that have been incoporated so far and y-axis represents an error rate.

In Table 3, we have listed parameter sets used in the proposed technique for a reproduction purpose. Note that numbers for τ are the relative values of the maximum of the selection score given in Eq. (66). Comparing parameters listed in Table 3 with Figure 13, where the number of classifiers that have to produce the class posterior probability of one for the selection score defined in Eq. (66) to be maximal is illustrated against different γs, it can be easily seen that unlabeled samples that the classifiers are maximally disagreeing (i.e. $P(y|x_i;\boldsymbol{\theta}^{(1)t}) = 1$ and $P(y|x_i;\boldsymbol{\theta}^{(2)t}) = 0.5$, or vice versa for $x_i \in \mathcal{U}^t$.) are preferred for both USPS and COIL-100 data sets. Nevertheless, one should not conclude this is indeed the case all the time. The risk thatwrong class labels are included might be different even with the same selection score depending on (a) classifiers used in an ensemble, and (b) data

Table 3: The list of chosen parameters

Data sets	USPS			COIL-100		
♯ of initial labeled data	5%	10%	20%	5%	10%	20%
γ	0.44	0.46	0.42	0.44	0.44	0.46
τ	0.88	0.88	0.84	0.84	0.84	0.88
η	4.8	4.8	2.4	4.8	4.8	4.8

sets tested as we will see in next section.

4.5.2 Comparisons between different sizes of an ensemble

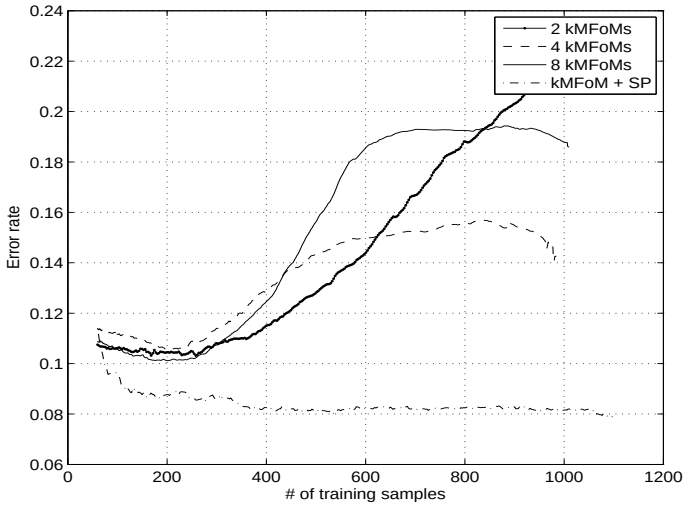
In this set of experiments, we examined how the proposed technique would work as the size of the ensemble was increased. To this end, we constructed an ensemble of classifiers consisting of kernelized MFoM classifiers, while varying the number of classifiers in the ensemble between two, four, and eight. For each kernelized MFoM classifier in the ensemble, we randomly chose a subset of training data to diversify classification outputs (i.e., we created different sets of $\mathcal{I}_s$. Please refer to Chapter 3 for more information about $\mathcal{I}_s$). Similar to the experiments in Section 4.5.1, we also created three different sizes of initially labeled data sets, such as 5%, 10%, and 20% of the entire training data set. As baseline systems, we prepared for two configurations:

- systems exploiting the same number of kernelized MFoM classifiers but generating S^t based on confidence scores only (*2kMFoM-Conf, 4kMFoM-Conf, and 8kMFoM-Conf*), and
- systems using a single kernelized MFoM classifier with a single NN-based classifier where S^t was created according to the selection score (*kMFoM+SP*).

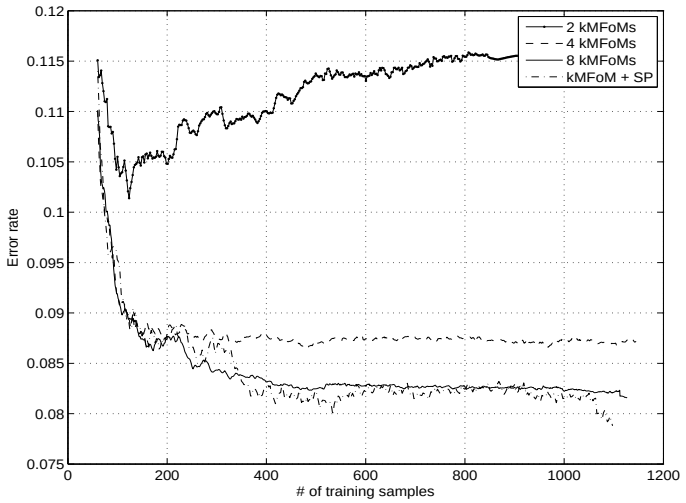
We also denote the proposed framework with different sizes of ensembles as *2kMFoM-EER, 4kMFoM-EER, and 8kMFoM-EER*, respectively. The purpose of the comparisons of the proposed framework with the first baseline systems is to verify the robustness of the proposed technique as well as to illustrate the characteristics of the technique in a function of the number of classifiers in an ensemble. On the other hand, the purpose of the second baseline system is to highlight the importance of the diversity of the outputs of the classifiers used in an ensemble.

Figures 16, 17, and 18 summarize the comparison results on the USPS data set. First, in Figure 16, performance comparison curves between the proposed framework and the baseline systems are shown where the size of initially labeled data set is set to 5%. In Figure 16-(a), the error rates of the confidence score based techniques (i.e., *2kMFoM-Conf, 4kMFoM-Conf, and 8kMFoM-Conf*) are given in the top three curves against the number of samples incorporated into the incremental learning procedures. The bottom curve represents the error rates corresponding to the baseline system *kMFoM+SP*. It is clearly seen that all of the three confidence score based techniques diverge after a few iterations mainly because of over-fitting to the training samples initially given. On the other hand, in Figure 16-(b), the error rates of the proposed technique (i.e., *2kMFoM-EER, 4kMFoM-EER, and 8kMFoM-EER*) at the top three curves as well as those of the baseline system *kMFoM+SP* are drawn in the bottom. It is shown that the proposed technique is able to enhance initial models except for the case when there are two classifiers in an ensemble. In fact, the more classifiers exist in an ensemble, the better the resulting models perform, as we expected in Section 4.3. Interestingly, the size of ensemble needs to be increased to eight to approach to the performance of the baseline *kMFoM+SP*, when an ensemble consists of kernelized MFoM classifiers only. This indicates an advantage of having diverse classification outputs, which has been a key message of the Co-training method.

Next, in Figure 17, classification error rates of the baseline systems and the proposed technique are drawn for a case when the size of initially labeled samples is 10% of the entire training data. Comparing the performance curves for the confidence score based techniques (i.e., *2kMFoM-Conf, 4kMFoM-Conf, and 8kMFoM-Conf*, the top three curves in Figure 19-(a)) and the other baseline system (i.e., *kMFoM+SP*, the bottom curve in Figure 19-(a)), it is clearly seen that confidence score based techniques are not able to retain the convergence property so that the final model performs worse than initial models. In Figure 17-(b), a similar tendency as in Figure 16-(b) is observed, where the systems with the proposed technique (i.e., *2kMFoM-EER, 4kMFoM-EER, and 8kMFoM-EER*) exhibit quite a bit of performance improvement over their corresponding initial models. One thing to note here is that the performance gap between different numbers of kernelized MFoM classifiers used

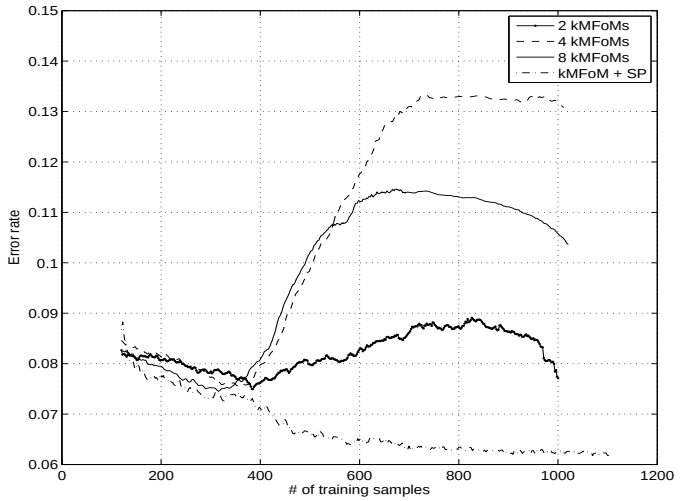


(a) Performance comparison curves for confidence score based methods with different sizes of ensembles such as 2, 4 ,and 8, and kMFoM+SP

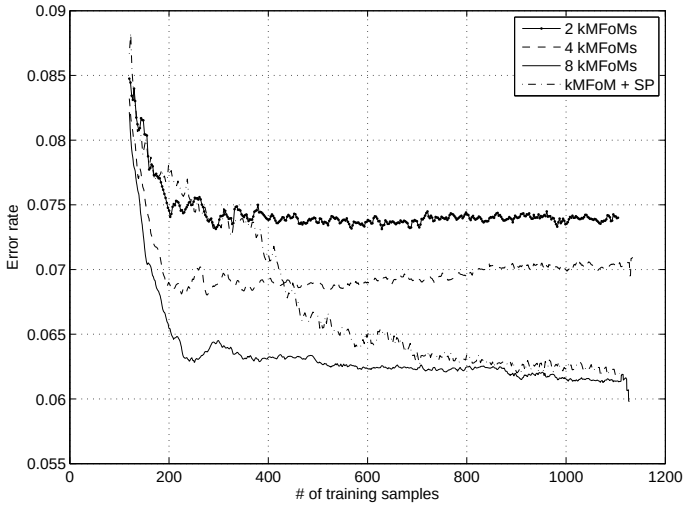


(b) Performance comparison curves for the proposed technique with different sizes of ensembles such as 2, 4, and 8, and kMFoM+SP

Figure 16: Performance comparison curves between the proposed technique (i.e. using the weighted combination of the expected error reduction and the confidence score to select unlabeled samples) and a confidence score based selection method when the number of classifiers are increased from 2 to 8. 5% of the training samples are chosen for the initial labeled set. x-axis represents the number of training samples that have been incoporated so far and y-axis represents an error rate. (a) Confidence score based technique only. (b) The proposed techniques. Note that n kMFoM means an ensemble of n kernelized MFoM classifiers. kMFoM + SP represents the performance of the proposed technique using two different classifiers, such as kernelized MFoM learning and a spectral clustering technique based NN classifier.



(a) Performance comparison curves for confidence score based methods with different sizes of ensembles such as 2, 4 ,and 8, and kMFoM+SP

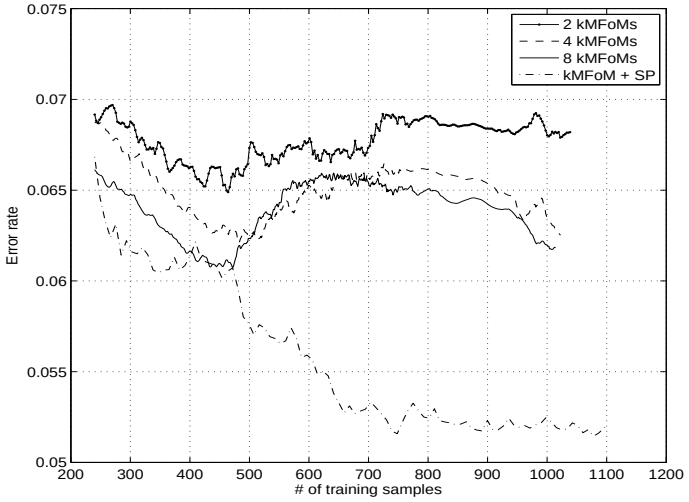


(b) Performance comparison curves for the proposed technique with different sizes of ensembles such as 2, 4, and 8, and kMFoM+SP

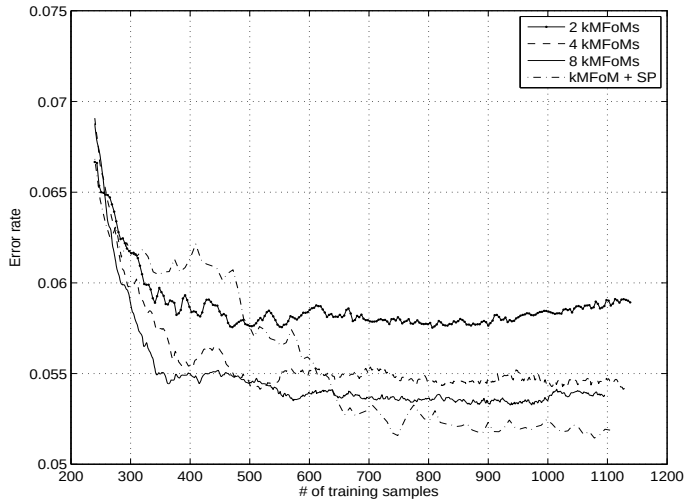
Figure 17: Performance comparison curves between the proposed technique and a confidence score based selection method on the USPS data set when the number of classifiers in an ensemble are increased from 2 to 8. 10% of the training samples are chosen for the initial labeled set. Refer to the descriptions in Figure 16 for other details

in an ensemble is narrowed. This implies that when we have better initial models, we might need fewer classifiers in an ensemble for estimating the expected error reduction.

Finally, Figure 18 depicts the performance comparison curves of the case when 20% of training data are used to learn initial models. As seen in Figure 18-(a), the systems with



(a) Performance comparison curves for confidence score based methods with different sizes of ensembles such as 2, 4 ,and 8, and kMFoM+SP



(b) Performance comparison curves for the proposed technique with different sizes of ensembles such as 2, 4, and 8, and kMFoM+SP

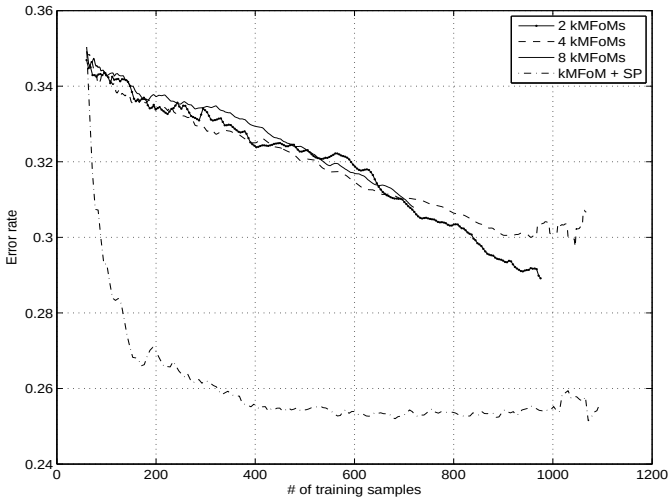
Figure 18: Performance comparison curves between the proposed technique and a confidence score based selection method on the USPS data set when the number of classifiers in an ensemble are increased from 2 to 8. 20% of the training samples are chosen for the initial labeled set. Refer to the descriptions in Figure 16 for other details

confidence score based techniques (i.e., *2kMFoM-Conf, 4kMFoM-Conf, and 8kMFoM-Conf*) are not diverged thank to good initial models, but they still fail to improve the initial models. What has happened is that, they enhance the classification models, initially, but start to fall out as the models are over-fitted to a small set of the existing training samples. On the

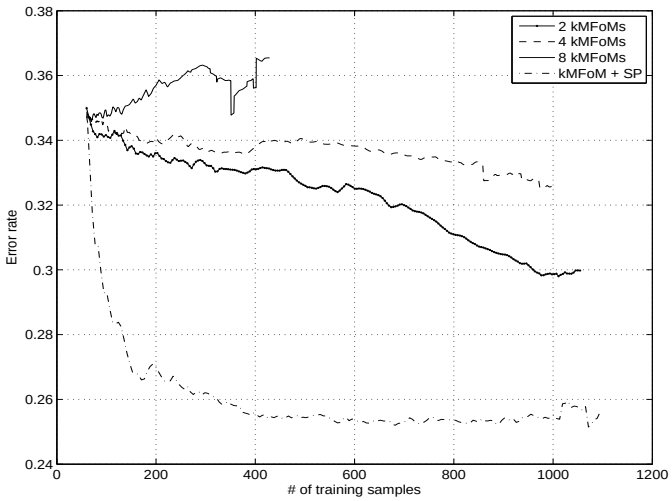
other hand, the systems with the proposed technique (i.e., *2kMFoM-EER, 4kMFoM-EER, and 8kMFoM-EER*) are able to show performance improvements consistently. It should be noted that in Figure 18-(b), the performance gaps measured between them are further narrowed from the case discussed in Figure 17-(b). In particular, even with an ensemble of four kMFoM classifiers (i.e., *4kMFoM-EER*), comparable classification error rates are shown for the best performing system, *kMFoM+SP*, where we have used distinct classifiers in the ensemble. This clearly demonstrates the power of having good initial models in a semi-supervised incremental learning framework.

Next, we illustrate comparison results on the COIL-100 data set. In particular, Figure 19 presents comparison results when 5% of training data have been used for building initial models. Comparing the top three curves in in Figure 19-(a) and those in Figure 19-(b), it is seen that the proposed technique is not able to demonstrate its effectiveness over the baseline systems with confidence score based techniques (i.e., *2kMFoM-Conf, 4kMFoM-Conf, and 8kMFoM-Conf*) unlike that observed in the USPS data set. When eight kernelized MFoM classifiers are used in an ensemble, the performance of the initial model even deteriorates. In fact, on the contrary to the USPS data set, in this case, the more classifiers are used, the worse performance becomes mainly because of a combination of two issues: (a) poor initial models, and (b) highly correlated outputs of classifiers in an ensemble. As for the first issue, note that the initially given error rate is over 30% as seen in Figure 19-(b). Furthermore, although the outputs of kernelized MFoM classifiers were randomized, they were based on the same training set and the same training algorithm, which created the highly correlated confidence scores and posterior probabilities across the classifiers in the ensemble. Thus, the more classifiers are used in estimating the expected error reduction, the more quickly classification models will be over-fitted to the incorrectly generated class labels.

In Figure 20, initial labeled samples are now increased to 10% of the entire training samples. Comparing Figure 20-(a) with -(b), it is clearly seen that the proposed technique still does not outperform the baseline systems based on confidence scores (i.e., *2kMFoM-Conf, 4kMFoM-Conf, and 8kMFoM-Conf*). As demonstrated in Figure 20-(b), there is a large gap in terms of error rates between the systems with the proposed technique (i.e.,



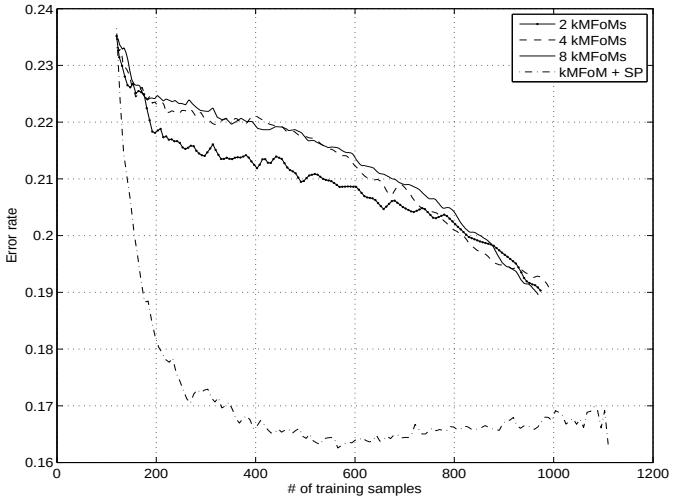
(a) Performance comparison curves for confidence score based methods with different sizes of ensembles such as 2, 4 ,and 8, and kMFoM+SP



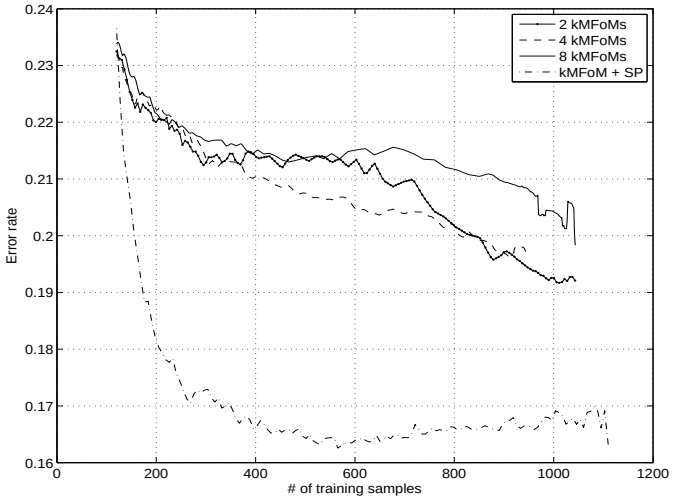
(b) Performance comparison curves for the proposed technique with different sizes of ensembles such as 2, 4, and 8, and kMFoM+SP

Figure 19: Performance comparison curves between the proposed technique and a confidence score based selection method on the COIL data set when the number of classifiers in an ensemble are increased from 2 to 8. 5% of the training samples are chosen for the initial labeled set. Refer to the descriptions in Figure 16 for other details

2kMFoM-EER, 4kMFoM-EER, and 8kMFoM-EER) and the system with the proposed technique using different classifiers (i.e., *kMFoM+SP*). Again, this confirms our previous claim that highly correlated prediction results restrict the ability to construct a *good* selection set $\mathcal{S}^t$ that preserves the convergence property of an incremental learning framework.



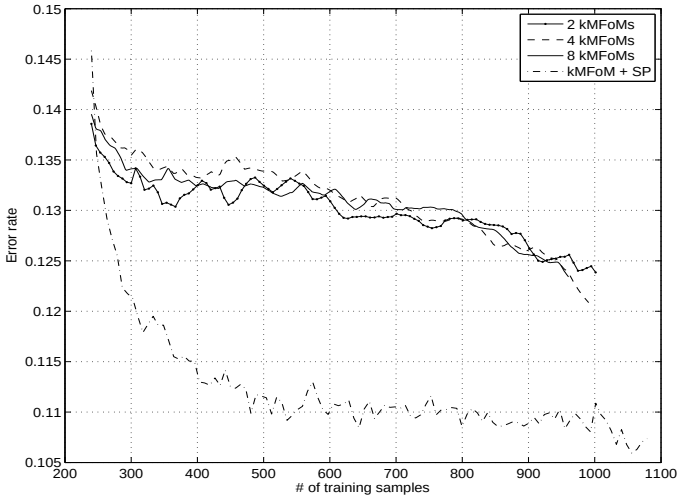
(a) Performance comparison curves for confidence score based methods with different sizes of ensembles such as 2, 4 ,and 8, and kMFoM+SP



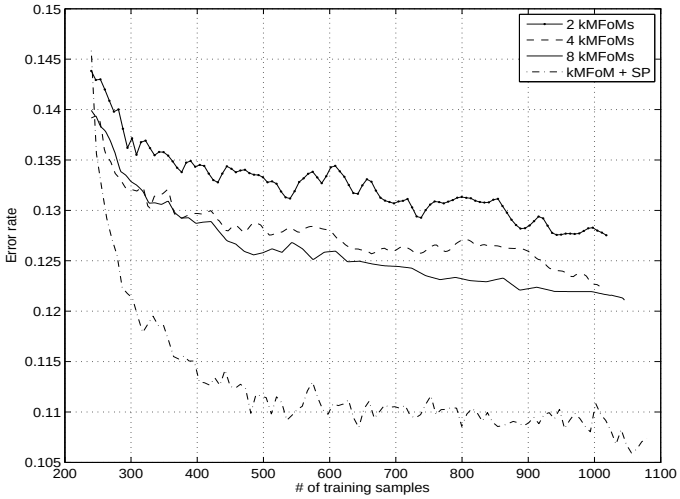
(b) Performance comparison curves for the proposed technique with different sizes of ensembles such as 2, 4, and 8, and kMFoM+SP

Figure 20: Performance comparison curves between the proposed technique and a confidence score based selection method on the COIL data set when the number of classifiers in an ensemble are increased from 2 to 8. 10% of the training samples are chosen for the initial labeled set. Refer to the descriptions in Figure 16 for other details

If we increase the size of initially labeled data set further to 20%, the proposed technique start to show one of its properties; a faster rate of the performance improvement compared to that of the baseline systems with the confidence score based methods. To see this, one can take a look at the middle part of the top three curves in Figure 21-(a) and those in



(a) Performance comparison curves for confidence score based methods with different sizes of ensembles such as 2, 4 ,and 8, and kMFoM+SP



(b) Performance comparison curves for the proposed technique with different sizes of ensembles such as 2, 4, and 8, and kMFoM+SP

Figure 21: Performance comparison curves between the proposed technique and a confidence score based selection method on the COIL data set when the number of classifiers in an ensemble are increased from 2 to 8. 20% of the training samples are chosen for the initial labeled set. Refer to the descriptions in Figure 16 for other details

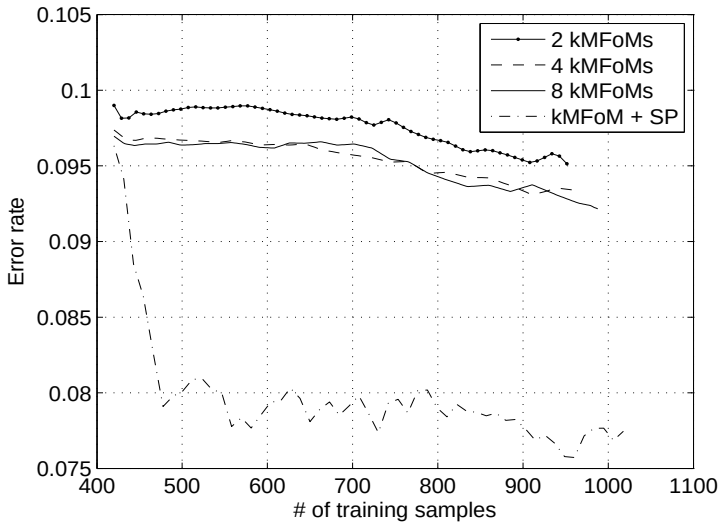
Figure 21-(b). Moreover, unlike previous cases when the initial label sets were 5% or 10%, the more classifiers are used in estimating the expected error reduction, the better the final model performs. These findings, in fact, highlight the importance of initial models to obtain the robustness of the proposed technique in a case where the outputs of the models are

highly correlated. In such cases, we might merely increase the risk of using incorrect class labels without having any advantages from an ensemble of classifiers.

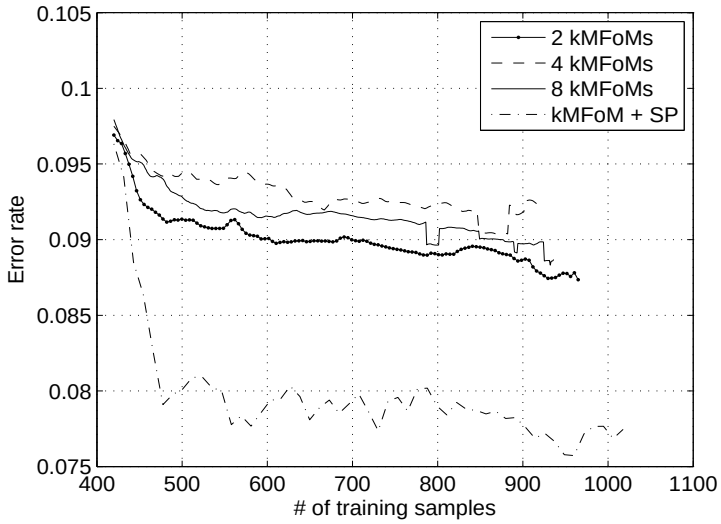
To see how much initial models should be improved for the proposed framework to clearly present its effectiveness in a highly correlated output case, an extra set of experiments was conducted in which the size of initial labeled set was raised to 35% of the entire training data. The baseline systems were configured in the same way as before. The comparison results are presented in Figure 22. As is clearly seen in Figure 22-(a) and -(b), the proposed framework now outperforms the confidence score based selection scheme in all cases (i.e., two, four, and eight kMFoM classifiers in an ensemble). Moreover, as also seen in the top three curves from both figures, the rate of the classification error reduction of the proposed technique is much faster than that of the confidence score based systems during the first few iterations. This again indicates the benefit of an expected error reduction function for semi-supervised incremental learning. Nevertheless, the use of multiple randomized kernelized MFoM classifiers is not as much effective as the case when two different classifiers are used in an ensemble; in Figure 22-(b), note the large gap between the bottom curve and the top three curves. This shows a clear advantage of exploiting different classification algorithms, which in general reduces the risk of over-fit and bias during estimation of the expected error reduction. We summarize some aspects of the proposed framework that have been revealed through a set of aforementioned experiments. First, the diversity of classifiers is important to reduce the risk of jeopardizing the convergence property. Second, if possible, better initial models should be pursued. We shall see a viable solution to achieve good initial models in Chapter 5. In Chapter 6 we will then come back to discuss how to integrate the technique described in Chapter 5 into the semi-supervised incremental learning framework proposed in this chapter.

4.6 Summary

This chapter mainly investigates a novel semi-supervised incremental learning framework that takes advantage of an error reduction measure, a quantity that measures the contribution of unlabeled samples on reducing classification errors if included into the learning



(a) Performance comparison curves for confidence score based methods with different sizes of ensembles such as 2, 4 ,and 8, and kMFoM+SP



(b) Performance comparison curves for the proposed technique with different sizes of ensembles such as 2, 4, and 8, and kMFoM+SP

Figure 22: Performance comparison curves between the proposed technique and a confidence score based selection method on the COIL data set when the number of classifiers in an ensemble are increased from 2 to 8. 35% of the training samples are chosen for the initial labeled set. Refer to the descriptions in Figure 16 for other details. Note that this time, the proposed technique outperforms the confidence score based selection method and the rate of performance increase is also much faster.

process. Given a small set of labeled and a large amount of unlabeled data, a handful set of unlabeled samples was chosen by jointly considering four key factors: (a) a confidence score, (b) an expected error reduction measure, (c) a class prior distribution, and

(d) a Zipf distribution. Then, the selected samples were used to update model parameter vectors. For robust estimation of the expected error reduction, an ensemble of classifiers was also exploited. Because no restriction has been imposed on a type of a classifier used, many existing learning algorithms can be integrated into the proposed learning framework, such as a spectral clustering technique based NN classifier, or kMFoM classifiers. Extensive experiments were performed on two real-word image data sets: (a) the USPS handwritten recognition data set, and (b) the COIL-100 object recognition data set. Experimental results revealed that the proposed technique outperformed two baseline semi-supervised incremental learning systems, namely a confidence score based method and the Co-training method. Performance comparisons between different sizes of ensembles showed interesting properties of the proposed technique. First, the more classifiers exist in an ensemble, the better performance is achieved in general. Second, diverse classifiers in the ensemble is preferable. Finally, the performance of an initial model is important to avoid over-fitting and to ensure the convergence of the incremental learning procedures. Note that in image concept modeling, the performance of an initial model can be improved by incorporating more features. In Chapter 5, we therefore discuss how to take advantage of multiple features and how to unify them into a single feature space given unlabeled samples.

Chapter V

AN AGREEMENT FUNCTION FOR MULTI-VIEW SEMI-SUPERVISED LEARNING (SSL)

In Chapter 4, we have seen that the performance of an initial model plays a key role in ensuring the success of a semi-supervised incremental learning framework. One natural approach to enhancing the performance of a classification model is to take advantage of multiple features that are complementary to each other. In the literature, there are mainly two different techniques, namely, early fusion and late fusion, depending on at what stage a classifier is trained. Specifically, in early fusion, we combine feature vectors together, creating a single, large feature vector before a classifier is learned. On the other hand, in late fusion, classifiers are trained first individually for each feature and then a meta-classifier is learned on top of the outputs of the classifiers. In Section 2.4, the pros and cons for each fusion method are explained in more detail. Briefly, in a semi-supervised setting, a late fusion approach tends to suffer from an over-fitting problem due to a small number of labeled data samples. In many cases, the decisions that individual classifiers make for labeled samples is either absolute yes or absolute no. Therefore, research on an early fusion approach has attracted much attention when a classification system is built with a small set of label data along with a large number of unlabeled samples.

One of the popular semi-supervised learning (SSL) frameworks capable of handling multiple features is a multi-view learning technique. Note that each view corresponds to each feature space. As discussed in Section 2.4, in multi-view learning, one usually exploits an agreement assumption stating that the models learned from individual features alone should agree in their class predictions. More formally, suppose there are multiple feature spaces, say $\mathcal{X}^{(1)}$ and $\mathcal{X}^{(2)}$, and the corresponding discriminant functions learned on $\mathcal{X}^{(1)}$ and $\mathcal{X}^{(2)}$,

$f^{(1)} : \mathcal{X}^{(1)} \times \mathcal{Y} \rightarrow \mathbb{R}$ and $f^{(2)} : \mathcal{X}^{(2)} \times \mathcal{Y} \rightarrow \mathbb{R}$, respectively. Then, $f^{(1)}$ and $f^{(2)}$ should satisfy

$$f^{(1)}(x^{(1)}, y; \boldsymbol{\theta}^{(1)}) = f^{(2)}(x^{(2)}, y; \boldsymbol{\theta}^{(2)}) \quad (74)$$
$$\text{for } \{(x^{(1)}, x^{(2)}) | P(x^{(1)}, x^{(2)}) > 0, x^{(1)} \in \mathcal{X}^{(1)}, x^{(2)} \in \mathcal{X}^{(2)}\} \text{ and } \forall y \in \mathcal{Y},$$

where $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$ are parameter vectors for $f^{(1)}$ and $f^{(2)}$, and the superscript (j) indicates an entity associated with the j^{th} feature space (e.g., $x^{(1)}$ is a feature vector in the first feature space $\mathcal{X}^{(1)}$, $f^{(2)}$ represents a discriminant function trained on feature vectors in the second feature space, etc.). Note that the superscript (j) in this chapter is different from that used in Chapter 4. In particular, the former indicates the j^{th} feature space, while the latter corresponds to the j^{th} classifier. However, they will be easily differentiable from the context because when the superscript is used to refer to a feature space, it will be applied to a feature vector x or a discriminant function f. If the superscript refers to a classifier, it will be used with a parameter vector, say $\boldsymbol{\theta}$.

In practice, Eq. (74) might be a too restrictive to be satisfied. Alternatively, techniques proposed in [90, 15, 92, 111] use a squared sum of differences between $f^{(1)}$ and $f^{(2)}$, referred to as co-regularization, to impose the agreement assumption as follows:

$$\sum_{i=1}^{N_u} \sum_{y \in \mathcal{Y}} \left[f^{(1)}(x_{u_i}^{(1)}, y; \boldsymbol{\theta}^{(1)}) - f^{(2)}(x_{u_i}^{(2)}, y; \boldsymbol{\theta}^{(2)}) \right]^2, \quad (75)$$

where the subscript u_i represent the i^{th} sample in an unlabeled data set $\mathcal{U}$, and N_u is the cardinality of $\mathcal{U}$. However, the use of Eq. (75) to enforce the agreement assumption might not always be beneficial to increase the prediction accuracy. For example, suppose we have an image spam email, an email that contains images filled with spam messages and texts that look completely legitimate. Suppose further that a spamfilter that classifies emails based on some image content analysis is given as in [20]. Then, according to the co-regularization term defined in Eq. (75), the class label for the textual part of an image spam should also be spam even if the content is non-spam. Now, a problem arises when the textual components of image spam are similar to those of legitimate emails. If this happens, the legitimate emails will also be labeled as spam, and eventually, the entire spam filtering system will be poisoned.

To tackle this issue, in this chapter, we define a specific type of noise, called disagreement noise, as follows: given multiple features, the disagreement noise occurs if more than two class labels are assigned to a single data sample depending on the feature to which a labeler refers (given this definition, it can be said that there exists disagreement noise in image spam emails because if the labeler perceives textual parts only, such image spam emails will be labeled as legitimate). Recently, a technique to deal with the disagreement noise in a multi-view learning framework has been proposed in [22]. In particular, multimedia data streams such as audio and video signals were considered where one of the signals can be blacked out due to a fault of the corresponding sensor. The technique proposed in [22] took care of such blackouts by assuming that there was a mixture model with three mixture components, each of which corresponded to the followings: audio, video, and blacked-out signals. However, in more general cases, this mixture modeling assumption might not fit well to many disagreement noises.

Instead, therefore, we investigate a technique to set up an agreement function, conveying our belief of the degree that models learned on each feature should agree in their class label predictions. Inspired by the analysis of the image spam email case, our algorithm of computing the agreement function depends upon the local structure of each feature space. More precisely, the value of an agreement function is proportional to the amount that the neighboring information of a certain unlabeled sample is shared across feature spaces (i.e., views). To test the effectiveness of the proposed agreement function, we incorporate it into the state-of-the-art co-regularized SSL algorithm originally presented in [92] by formulating a closed form solution for a kernel function with the agreement function, which in turn, unifies multiple views into a single reproducing kernel Hilbert space (RKHS). Experimental results on artificially generated data sets showed that our technique outperformed the original co-regularized SSL technique whenever disagreement noise exists. Additional experiments on the TREC05 spam corpus also revealed that first, there was indeed disagreement noise on image spam emails and second, the use of an agreement function further reduced the classification error over the conventional co-regularized SSL algorithms.

The remainder of this chapter is organized as follows: in Section 5.1 we present our

algorithms to compute the agreement function. In Section 5.2, we discuss a multi-view SSL framework to which the agreement function is incorporated while presenting a mathematical formulation for the kernel function that combines multiple features spaces into a single RKHS. Sections 5.3 and 5.4 then present experimental results on artificially generated data sets and on the TREC05 spam corpus, respectively. Finally, Section 5.5 summarizes this chapter with some concluding remarks.

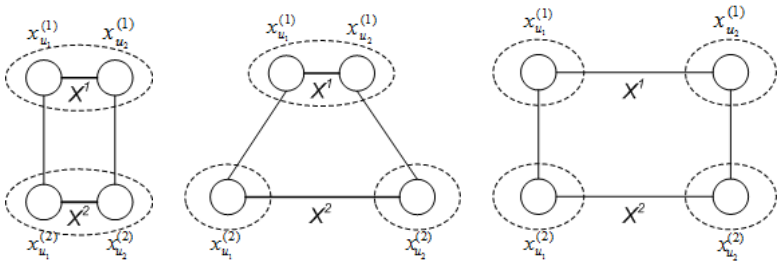
5.1 Algorithms to learn the agreement function

In this following, we describe an algorithm to evaluate an agreement function. In Section 5.1.1, we elaborate the discussion about the disagreement noise by providing a toy example using graphical representations. Extending the discussions in Section 5.1.1, a graph-based algorithm to compute an agreement function is presented in Section 5.1.2 followed by a probabilistic generative modeling approach in evaluating the agreement function in Section 5.1.3.

5.1.1 Graphical representations of the disagreement noise

Figure 23 illustrates simple graphical representations of how unlabeled samples with multiple views can be positioned. Here, we assume that there are two unlabeled samples with two views, say $x_{u_1} = (x_{u_1}^{(1)}, x_{u_1}^{(2)})$, $x_{u_2} = (x_{u_2}^{(1)}, x_{u_2}^{(2)})$, where each node in the graphs correspond to each of the feature vectors extracted from each view as seen in Figure 23. We also assume that there are two different type edges in Figure 23 as well: (a) edges connecting nodes within a view, and (b) edges connecting nodes across views. A dotted circle in Figure 23 is referred to as a neighboring set for a certain node such that the class labels for the samples in the neighboring set are assumed to be the same as the label of the corresponding node.

Among the three graphs in Figure 23 (i.e., G_1, G_2, G_3), the most interesting case is G_2 because the disagreement noise defined at the beginning of Chapter 5 will compromise a classification performance the most compared to the other configurations in Figure 23. To see this, suppose we have a binary-class problem, where the class labels are denoted as 0 and 1. Suppose further that the nodes corresponding to $x_{u_1}^{(1)}$ and $x_{u_1}^{(2)}$ are labeled as 0, and the node that represents $x_{u_2}^{(2)}$ has a class label of 1 as shown in Figure 24. Then, it can



(a) G_1 (b) G_2 (c) G_3

Figure 23: Simple graphical representations of possible configurations of two unlabeled samples with two views. The dotted circle represents a neighboring set of each node. The class labels of the samples in the neighboring set are assumed to be the same as that of each node. (a) Samples shares the same neighboring set consistently for both views. (b) Samples are in the same neighboring set for only one view. (c) Samples have distinct neighboring sets for both views.

be seen that based on the agreement assumption, $x_{u_2}^{(1)}$ should be assigned to class 1, but then this labeling scheme violates our premise regarding an neighboring set (i.e., samples in the same neighboring set should share the same class labels.). On the other hand, if we force $x_{u_2}^{(1)}$ to be classified as class 1, according to the labeling rule of a neighboring set, $x_{u_1}^{(1)}$ will be labeled as class 1, and subsequently, $x_{u_1}^{(2)}$ will also be assigned to class 1, which will hurt the classification accuracy. In Figure 24, there are two possible ways to deal with

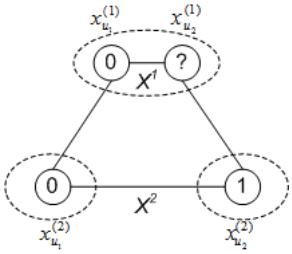


Figure 24: A graphical representation of what will happen with a presence of the disagreement noise. Note that the class labels for $x_{u_1}^{(1)}$, $x_{u_1}^{(2)}$, and $x_{u_2}^{(2)}$ are assumed to be class 0, class 0, and class 1, respectively. With this setup, it is not clear how to label $x_{u_2}^{(1)}$ as the agreement assumption says it should be classified as class 1, while the assumption regarding neighboring sets claims it should be labeled as class 0.

this disagreement noise: (a) to shrink the scope of the neighboring set of $x_{u_1}^{(1)}$ and $x_{u_2}^{(1)}$ so that the configuration of the nodes can be effectively changed from G_2 to G_3, or (b) to reduce the amount of agreement assumption between views. However, the first approach is, in general, infeasible because the size of the neighboring set is an inherent characteristic

of a certain data set. For a better understanding, consider an extreme case when the size of the neighboring set is one. In this case, samples in a certain data set are deemed to be independent to each other, so no helpful information can be extracted from unlabeled samples in estimating class boundaries. As an alternative, therefore, we focus on learning an agreement function with which we adjust selectively the level of agreement to be enforced for each unlabeled sample.

5.1.2 A graph-based algorithm to learn the agreement function

To develop an algorithm to learn the agreement function, let J be the total number of views. Then, the goal is to find an agreement function $\xi : \mathcal{U} \rightarrow \mathbb{R}$ from $N = N_l + N_u$ samples in $\mathcal{D} = \mathcal{L} \cup \mathcal{U}$ such that $\xi(x_{u_i})$ for $x_{u_i} = (x_{u_i}^{(1)} \dots x_{u_i}^{(J)})$ denotes the significance of agreement for the i^{th} unlabeled data sample in $\mathcal{U}$, where N_l and N_u are the number of labeled and unlabeled samples in $\mathcal{L}$ and $\mathcal{U}$, respectively. Note that for the notational simplicity, we use a shorthand notation for the agreement function as $\xi_{u_i} = \xi(x_{u_i})$ hereafter. The development of the agreement function can be started from the analysis of the disagreement noise in Figure 23. Specifically, in Figure 23-(b), half of the neighboring set in $\mathcal{X}^{(1)}$ coincides with that of $\mathcal{X}^{(2)}$, while in Figures 23-(a) and -(c), the neighboring sets in $\mathcal{X}^{(1)}$ are matched perfectly with those in $\mathcal{X}^{(2)}$. Based on this observation, we claim that the value of an agreement function should be proportional to the amount of neighboring information shared among different views. To quantify this idea, we first count the number of neighboring samples of x_{u_i} that repeatedly occur in all views, which we denote it as $\mathcal{N}_{x_{u_i}}$. More formally, let $\mathcal{N}_{x_{u_i}^{(j)}}$ be the neighboring set of x_{u_i} based only on the j^{th} view. Then, $\mathcal{N}_{x_{u_i}}$ is given by

$$\mathcal{N}_{x_{u_i}} = \sum_{m=1}^{N} \prod_{j=1 \dots J} I(x_m^{(j)} \in \mathcal{N}_{x_{u_i}^{(j)}}), \tag{76}$$

where $I(\cdot)$ is an indicator function. Note that here, a plain subscript m is used for representing a sample in the entire training data $\mathcal{D}$, while u_i is for a sample in an unlabeled data set $\mathcal{U}$ only. Thus, the summation in Eq. (76) is over $\mathcal{D}$, not just over $\mathcal{U}$. Intuitively, Eq. (76) can be interpreted as the size of the intersection of the neighboring sets of $x_{u_i}^{(j)}$ for $1 \leq j \leq J$ when the sets are projected onto the same space. Given Eq. (76), ξ_i can now be

defined as the normalized version of $\mathcal{N}_{x_{u_i}}$ by the size of the neighboring set for each view $\mathcal{N}_{x_{u_i}^{(j)}}$as follows:

$$\xi_i = \sum_{m=1}^{N} \prod_{j=1,\ldots,J} \frac{I(x_m^{(j)} \in \mathcal{N}_{x_{u_i}^{(j)}})}{|\mathcal{N}_{x_{u_i}^{(j)}}|}, \tag{77}$$

where $|\cdot|$ denotes a cardinality of a certain set.

A remaining question is how to find the neighboring set $\mathcal{N}_{x_{u_i}^{(j)}}$ for the unlabeled sample x_{u_i} given the j^{th} feature space. In this work, we use an affinity matrix $W^{(j)} \in \mathbb{R}^{N \times N}$ that encodes pairwise distances among the samples in $\mathcal{D}$. In particular, the element of $W^{(j)}$ at the m^{th} row and the n^{th} column, $\{W^{(j)}\}_{mn}$, is defined as $\{W^{(j)}\}_{mn} = e^{-\frac{1}{2h^2} d(x_m^{(j)}, x_n^{(j)})}$, if either $x_m^{(j)}$ is one of the q nearest neighbors (NNs) of $x_n^{(j)}$, or vice versa. If neither $x_m^{(j)}$ nor $x_n^{(j)}$ is one of the q NNs of each other, $\{W^{(j)}\}_{mn} = 0$. Note that $e^{-\frac{1}{2h^2} d(x_m^{(j)}, x_n^{(j)})}$ is an RBF kernel with a bandwidth parameter h. For $d(x_m^{(j)}, x_n^{(j)})$, a certain distance metric (e.g., an Euclidean distance or a cosine distance) between $x_m^{(j)}$ and $x_n^{(j)}$ can be used. Denoting $\{W^{(j)}\}_{mn}$ as $w_{mn}^{(j)}$, we can restate Eq. (77) for ξ_i as follows:

$$\xi_i = \sum_{m=1}^{N} \prod_{j=1,\ldots,J} \frac{w_{im}^{(j)}}{\sum_{n=1}^{N} w_{in}^{(j)}}. \tag{78}$$

Eq. (78) can be simplified further by using a matrix form. Let $D^{(j)} \in \mathbb{R}^{N \times N}$ be a diagonal matrix, each of the diagonal entries is the column-wise sum of a row in $W^{(j)}$. Using the Schure product (component-wise multiplication) denoted by $\circ$, let $W = W^{(1)} \circ \cdots \circ W^{(J)}$ and $D = D^{(1)} \circ \cdots \circ D^{(J)}$. Then, Eq. (78) can be written as

$$\begin{bmatrix} \boldsymbol{\zeta} \\ \boldsymbol{\xi} \end{bmatrix} = D^{-1} W \mathbf{1}, \tag{79}$$

where $\boldsymbol{\zeta}$ is an N_l dimensional vector that represents the values of an agreement function for the labeled samples in $\mathcal{L}$ and $\boldsymbol{\xi}$ is a vector form of ξ_i as in $\boldsymbol{\xi} = [\xi_i, \ldots, \xi_{N_u}]$ for the unlabeled samples in $\mathcal{U}$. $\mathbf{1}$ is an N dimensional vector filled with ones.

5.1.3 A probabilistic algorithm to learn the agreement function

The algorithm presented above can also be interpreted probabilistically using a generative process. Suppose $P(x)$ is the marginal probability distribution of x defined on $\mathcal{X}$, where

$\mathcal{X} = \mathcal{X}^{(1)} \times \mathcal{X}^{(2)} \times \cdots \times \mathcal{X}^{(J)}$ with a typical element $x = (x^{(1)}, x^{(2)}, \ldots, x^{(J)})$. Eventually, we will show that the agreement function ξ_i is equal to $P(x_{u_i})$, the probability of observing x_{u_i} according to the generative process. To derive this equality, let us first assume there is a latent random variable L such that L is a process to generate a particular data sample. Given L, let us assume $P(x^{(j)}|L)$, a probability of observing a feature vector $x^{(j)}$ in $\mathcal{X}^{(j)}$ given L, that models the randomness in measuring feature vectors in $\mathcal{X}^{(j)}$. Since these noises can be assumed to be independent among different views, $P(x^{(j)}|J)$ can also be assumed to be conditionally independent for $1 \leq j \leq J$ given L. Then, $P(x)$ can be written as

$$P(x) = \sum_{L} \prod_{j=1,\ldots,J} P(x^{(j)}|L)P(L). \tag{80}$$

Now, suppose $P(x^{(j)}|L = l)$ is approximated with $P(x^{(j)}|x_l^{(j)})$. Using a classical kernel density estimation technique [71], $P(x_i^{(j)}|x_l^{(j)})$, a probability of observing $x_i^{(j)}$ given $x_l^{(j)}$ can then be modeled as

$$P(x_i^{(j)}|x_l^{(j)}) = \frac{k(x_i^{(j)}, x_l^{(j)})}{\sum_{m=1}^{N} k(x_m^{(j)}, x_l^{(j)})}, \tag{81}$$

where $k(x_m^{(j)}, x_l^{(j)})$ is a kernel function, which can be defined as $e^{-\frac{1}{2h^2} d(x_m^{(j)}, x_l^{(j)})}$ given a certain distance metric $d(x_m^{(j)}, x_l^{(j)})$ between $x_m^{(j)}$ and $x_l^{(j)}$, and a bandwidth parameter h. Note that $P(x_i^{(j)}|x_l^{(j)})$ given by Eq. (81) can be viewed as the transition probability from $x_l^{(j)}$ to $x_i^{(j)}$ in a Markov random walk with a transition matrix $P^{(j)} \in \mathbb{R}^{N \times N}$, where $\{P^{(j)}\}_{li} = P(x_i^{(j)}|x_l^{(j)})$. Assuming we have a uniform distribution for $P(L)$, Eq. (80) can then be simplified as

$$\begin{aligned} \boldsymbol{p} &= \frac{1}{N} \cdot \mathbf{1}^T (P^{(1)} \circ \cdots \circ P^{(J)}) && (82) \\ &= \frac{1}{N} \cdot \mathbf{1}^T D^{-1} W && (83) \\ &= \tilde{\boldsymbol{\xi}}, && (84) \end{aligned}$$

where $\boldsymbol{p}$ is a combination of a vector form of $P(x_{l_i})$, the marginal probability of the i^{th} sample in $\mathcal{L}$ and that of $P(x_{u_i})$ evaluated on $\mathcal{U}$. Additionally, $W = W^{(1)} \circ \cdots \circ W^{(J)}$, where $\{W^{(j)}\}_{li} = k(x_i^{(j)}, x_l^{(j)})$, and $D = D^{(1)} \circ \cdots \circ D^{(J)}$, where $D^{(j)}$ is a diagonal matrix such that its i^{th} diagonal entry is given by $\sum_{m=1}^{N} k(x_i^{(v)}, x_m^{(v)})$, respectively. Then, one can easily see the similarity between the definitions of the agreement function given in Eq. (83) and Eq.

(79). Therefore, $P(x_{u_i})$ can be used as an alternative definition of the agreement function. The main difference between Eqs. (83) and (79) is how the unit vector $\mathbf{1}$ is multiplied. In particular, $\mathbf{1}$ is applied column-wise in Eq. (79) while in Eq. (83), the vector is multiplied row-wise. Nevertheless, the equality relations can still be derived by using a symmetric matrix, such as $D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$, instead of $D^{-1}W$ in Eqs. (79) and (83).

5.2 A multi-view SSL framework with an agreement function

A multi-view SSL algorithm with an agreement function, ξ, is implemented by extending a co-regularized learning framework proposed in [92]. For simplicity, suppose we have $J = 2$. Then, we need to solve the following optimization problem.

$$\min_{\substack{f=f^{(1)}+f^{(2)} \\ f^{(1)}\in\mathcal{H}_K^{(1)}, f^{(2)}\in\mathcal{H}_K^{(2)}}} \sum_{y\in\mathcal{Y}} \{\lambda_1 \|f^{(1)}(\cdot,y)\|^2_{\mathcal{H}_K^{(1)}} + \lambda_2 \|f^{(2)}(\cdot,y)\|^2_{\mathcal{H}_K^{(2)}} + \mu \sum_{i\in\mathcal{U}} \xi_i [f^{(1)}(x^{(1)}_{u_i},y) - f^{(2)}(x^{(2)}_{u_i},y)]^2\} + \frac{1}{N_l}\sum_{i=1}^{N_l} V(\delta(x_{l_i},y_{l_i}),y_{l_i}), \quad (85)$$

where $f^{(1)}(x^{(1)},y)$ and $f^{(2)}(x^{(2)},y)$ is a shorthand notation of $f^{(1)}(x^{(1)},y;\boldsymbol{\theta}^{(1)})$ and $f^{(2)}(x^{(2)},y;\boldsymbol{\theta}^{(2)})$, respectively. Moreover, $\delta : \mathcal{X} \to \mathcal{Y}$ is a decision rule given by

$$\delta(x;\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)},\boldsymbol{\theta}^{(2)})) = \{y | f(x,y;\boldsymbol{\theta}) > 0, y \in \mathcal{Y}\}, \quad (86)$$

and $\mathcal{H}_K^{(1)}$ and $\mathcal{H}_K^{(2)}$ are RKHS's defined on $\mathcal{X}^{(1)}$ and $\mathcal{X}^{(2)}$, respectively. The term given by

$$\sum_{i\in\mathcal{U}} \xi_i \left[f^{(1)}(x^{(1)}_{u_i},y) - f^{(2)}(x^{(2)}_{u_i},y)\right]^2 \quad (87)$$

evaluates the co-regularization term on unlabeled samples with an agreement function ξ_i and $\frac{1}{N_l}\sum_{i=1}^{N_l} V(\delta(x_{l_i};\boldsymbol{\theta}),y_{l_i})$ is an empirical error evaluated over labeled samples. λ_1, λ_2, and μ are parameters to balance quantities among the norms in RKHS's $\mathcal{H}_K^{(1)}$ and $\mathcal{H}_K^{(2)}$, the co-regularization term, and the empirical error.

To solve the above optimization problem, we modify Theorem 2.2 in [92] to embed an agreement function ξ as follows:

Theorem 3. *Let be $\mathcal{H}_K^{(1)}$ and $\mathcal{H}_K^{(2)}$ are RKHS's and $k^{(1)}$ and $k^{(2)}$ be the corresponding reproducing kernels. Consider also a co-regularized objective function as in Eq. 85. Then,*

there exists an inner product on $\mathcal{H}_K$*, given by* $\mathcal{H}_K = \{f : f = f^{(1)} + f^{(2)}, f^{(1)} \in \mathcal{H}_K^{(1)}, f^{(2)} \in \mathcal{H}_K^{(2)}\}$*, for which* $\mathcal{H}_K$ *is an RKHS with norm defined as*

$$\|f\|_{\mathcal{H}_K}^2 = \min_{\substack{f=f^{(1)}+f^{(2)} \\ f^{(1)}\in\mathcal{H}_K^{(1)}, f^{(2)}\in\mathcal{H}_K^{(2)}}} \lambda_1 \|f^{(1)}(\cdot, y)\|_{\mathcal{H}_K^{(1)}}^2 + \lambda_2 \|f^{(2)}(\cdot, y)\|_{\mathcal{H}_K^{(2)}}^2 + \mu \sum_{i\in\mathcal{U}} \xi_i [f^{(1)}(x_{u_i}^{(1)}, y) - f^{(2)}(x_{u_i}^{(2)}, y)]^2 \quad (88)$$

and reproducing kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *given by*

$$k(x, z) = \lambda_1^{-1} k^{(1)}(x, z) + \lambda_2^{-1} k^{(2)}(x, z) - \mu \mathbf{d}_{\mathrm{x}}^{\mathbf{T}} \tilde{\mathrm{H}} \mathbf{d}_{\mathrm{z}}, \quad (89)$$

where $\tilde{H} = (\Xi^{-1} + \mu Q)^{-1}$ *such that* Ξ *is defined as a diagonal matrix of which the* i^{th} *diagonal entry is* ξ_i *and* $\{Q\}_{ij} = \lambda_1^{-1} k^{(1)}(x_{u_i}, x_{u_j}) + \lambda_2^{-1} k^{(2)}(x_{u_i}, x_{u_j})$*,* $i \in \mathcal{U}$*,* $j \in \mathcal{U}$*. Moreover* $\mathbf{d}_{\mathrm{x}}$ *and* $\mathbf{d}_{\mathrm{z}}$ *represent vectors such that* $\mathbf{d}_{\mathrm{x}} = [\lambda_1^{-1} k^{(1)}(x, x_{u_i}) - \lambda_2^{-1} k^{(2)}(x, x_{u_i}), i \in \mathcal{U}]^T$ *and* $\mathbf{d}_{\mathrm{z}} = [\lambda_1^{-1} k^{(1)}(x_{u_i}, z) - \lambda_2^{-1} k^{(2)}(x_{u_i}, z), i \in \mathcal{U}]$*, respectively.*

We have provided the proof of the above theorem in the Appendix.

Comparing Theorem 2.2 in [92] with Theorem 3, it can be seen that the main difference lies in how to construct the matrix $\tilde{H}$. Because $\mathbf{d}_{\mathrm{x}}$ and $\mathbf{d}_z$ can be viewed as vectors measuring the amount of disagreement between views, in Theorem 3, it can be considered that whether to enforce or to ignore disagreement noise for each unlabeled sample is determined by $\tilde{H}$. Additionally, Theorem 3 provides a simple method to solve Eq. (85). In particular, since we know the closed-form solution for a kernel function for the newly constructed RKHS $\mathcal{H}_K$, the norms in RKHS's $\mathcal{H}_K^{(1)}$ and $\mathcal{H}_K^{(2)}$ and the co-regularization term in Eq. (85) can be simplified into a single norm in $\mathcal{H}_K$. As a result, to solve Eq. (85) is equivalent to solve the following equations.

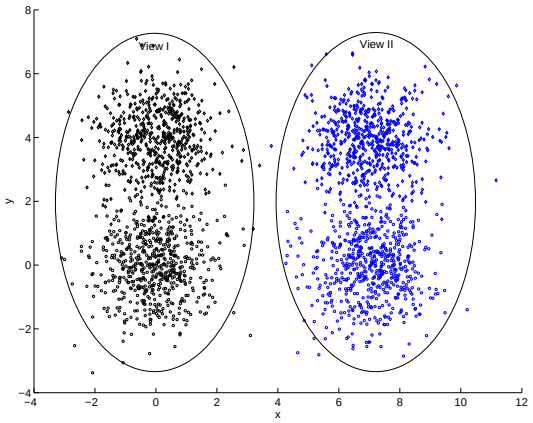
$$\min_{f\in\mathcal{H}_K} \frac{1}{N_l} \sum_{i=1}^{N} V(\delta(x_{l_i}, \boldsymbol{\theta}), y_{l_i}) + \lambda \sum_{y\in\mathcal{Y}} \|f(\cdot, y)\|_{\mathcal{H}_K}^2 , \quad (90)$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)})$. It can be seen that Eq. (90) is a special case of discriminative learning as discussed in Section 2.1; see Eq. (3). Therefore, we can apply any kernel based learning algorithms, such as a kernelized MFoM learning approach presented in Chapter 3 or other techniques like SVMs, GPs, etc., to solve Eq. (90).

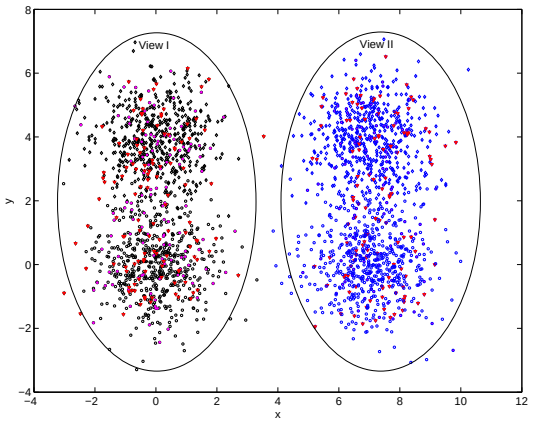
5.3 Multi-view SSL on artificial data sets

We first evaluated the proposed multi-view SSL framework on artificially generated data sets to test the effectiveness of the agreement function with a presence of the disagreement noise. The artificial data were generated using a mixture of two dimensional Gaussian distributions with four mixture components.

As shown in Figure 25-(a), a multi-view environment was created by tying two mixture components each, resulting in two distinct views. Each view was then assumed to have two

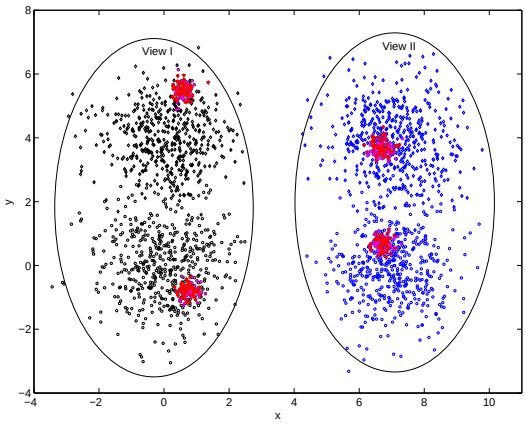


(a) A clean data set



(b) A data set with disagreement noise. Samples with disagreement noise are in magenta. Here, 20 percent of training data samples have been corrupted by the disagreement noise. For such samples, different class labels are assigned for different views.

Figure 25
Continued..



(c) A data set with disagreement noise. Samples with disagreement noise are in magenta. Similar to the data samples in (b), 20 percent of training data have been corrupted by disagreement noise. However, the area where the disagreement noise can occur is restricted to 0.2σ, where σ is the variance of the original Gaussian mixture component.

Figure 25 continued: An illustration of artificial data sets used to demonstrate the effectiveness of the agreement function in multi-view SSL. Samples are generated from a mixture of Gaussian distribution with four components where the means of the mixture components are adjusted to generate the Bayes error rate of 5 percent when no disagreement noise exists (i.e. in (a)). The variances for all four mixture components are set to unity. In (b), 20 percent of the data samples are compromised with the disagreement noise where the affected samples are colored in magenta. In (c), the same amount of the disagreement noise as the case in (b) is introduced, but the area where the noise is observed is much smaller.

classes, say, class -1 and class 1. The mixture component in the bottom was labeled as class -1, and the mixture component at the top was assigned to class 1. The variances of all mixture components were set to 1 while the means of the mixture components were adjusted to have a theoretical lower-bound for an error rate of 5% when no disagreement noise was assumed. In particular, means vectors were located at $[0, 0]$, $[0, 4]$, $[7, 0]$, and $[7, 4]$, respectively.

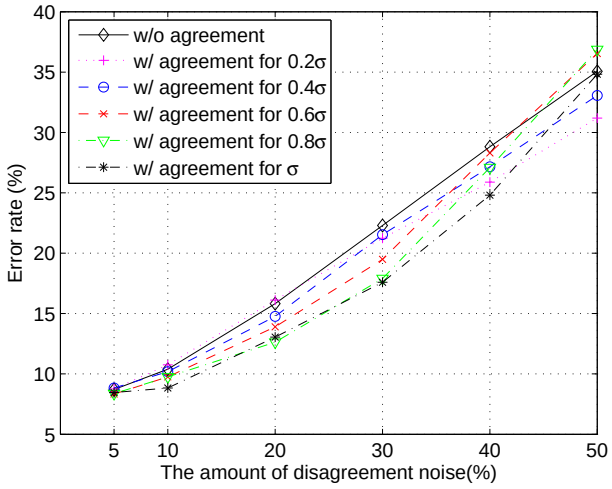
From each mixture component, we randomly generated 600 samples. Out of 600 samples, a handful number of samples (e.g., 20, 50) from each view were chosen for a labeled set. From the remaining data, we randomly chose 20% of them for testing. Disagreement noise was created by performing cross-pairing such that samples in the lower-left component were paired with samples in the upper-right component. Likewise, samples in the upper-left component were paired with samples in the lower-right component. Figure 25-(b) illustrates

an example of the resulting data set after this cross-paring where 240 out of 1200 (600 times two) pairs of samples have been corrupted by the disagreement noise.

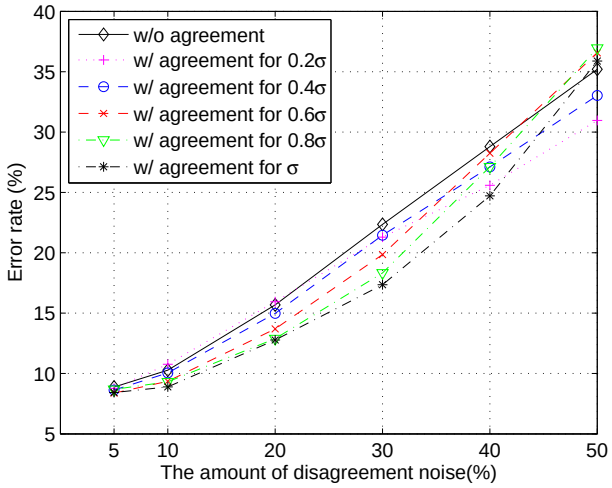
On the other hand, the disagreement noise might only occur from a small region of the feature space as shown in Figure 25-(c). To control the size of such a region, another Gaussian distribution whose variance was a fraction of the variance of the original mixture component (i.e., the mixture model from which data samples were generated.) was used. For example, let the variance of the original mixture component be σ. Then, in Figure 25-(c), we drew the samples that had been compromised with the disagreement noise from an extra Gaussian distribution whose variance was 0.2σ. Such samples are colored in magenta in Figure 25-(c). Note that the mean of the extra Gaussian distribution was determined randomly in a way that samples with the disagreement noise would have the variance of σ on average.

The baseline system was a co-regularized multi-view SSL implemented based on [92]. For fair comparison, we matched all setups except for λ_1, λ_2. For λ_1, λ_2, the best set of parameters for each case was determined through a cross-validation within a range of $1e^{-7}$ to $1e^{-2}$, incremented by a factor of 10. It should be noted that to compute Ξ, two more parameters needed to be set: (a) the number of nearest neighbors q and (b) the size of the bandwidth of an RBF kernel h. In this work, to lessen the efforts for parameter adjustment, q and h were simply set to 10 and an average pairwise Euclidean distance between training samples throughout experiments, respectively. Ξ was computed based on Eq. (82) with an exponent parameter $\eta \geq 0$ such that $\tilde{\boldsymbol{\xi}} = \boldsymbol{p}^{\eta}$, where $\boldsymbol{p}^{\eta}$ was a vector obtained from $\boldsymbol{p}$ with which each element of $\boldsymbol{p}$ was raised to the power of η. Therefore, by setting $\eta = 0$, our proposed algorithm would be reduced to the baseline system. Again, to minimize the efforts for parameter adjustment, η was simply set to 4.5. As for a classification model, a Regularized Least Square Regression (RLSR) algorithm, discussed in [48], was used due to its simplicity.

Figure 26 shows performance comparison curves of the proposed technique and the baseline system when the number of labeled samples is 20. Performances are reported in terms of an error rate after taking an average of the results of 100 runs for each of the following



(a) Performance comparison curves on the unlabeled set. The black solid line corresponds to the baseline system, while dotted lines represent the error rates of the proposed multi-view SSL framework.

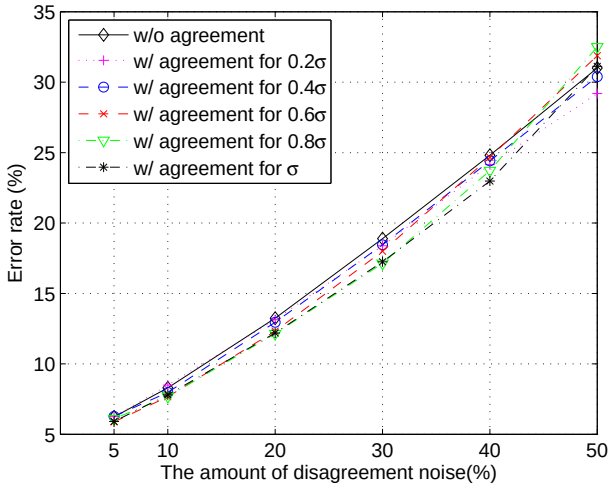


(b) Performance comparison curves on the test set. The black solid line corresponds to the baseline system, while dotted lines represent the error rates of the proposed multi-view SSL framework.

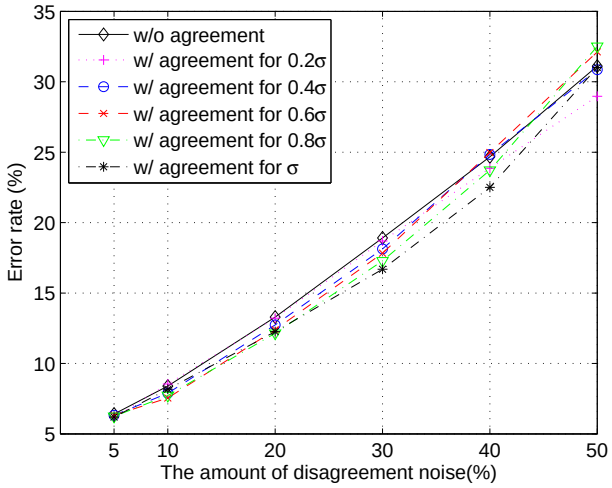
Figure 26: Performance comparison graphs between the baseline system and the proposed multi-view SSL technique when the number of labeled samples is set to 20. The solid line corresponds to the baseline system that does not use the agreement function. The dotted lines are for the proposed framework. x-axis represents the amount of disagreement noise, while y-axis is an error rate. For the unlabeled data set (i.e., illustrated in (a)), as the amount of disagreement noise and the area in feature spaces where the noise can occur increase, the proposed framework becomes more and more effective. The maximum performance gain is seen when the amount of the disagreement noise is 30 percent. For the test data set shown in (b), the same observations as the case with the unlabeled data set can be made.

cases: (a) varying the level of disagreement noise from 5% to 50%, and (b) varying the size of the region where disagreement noise occurs from 0.2σ to σ, where σ is the variance of the original mixture component. In Figure 26-(a), it can be clearly seen that the proposed technique (the dotted lines) outperforms the baseline system (the solid line) except when the level of disagreement noise is 50%. In particular, when we have the amount of disagreement noise of 20%, we achieve 17.5% of relative error rate reduction. On the other hand, when the level of disagreement noise is small, say 5%, the relative error rate reduction is 2.8%. Therefore, it can be said that the amount of disagreement noise is an important factor to determine the effectiveness of the proposed technique. The fact that the performances are degraded when the amount of the noise is 50% can be justified by the fact that there is no need to exploit co-regularization in the first place if half of the training samples have been corrupted by the disagreement noise. As shown in Figure 26-(b), the size of the compromised region in the feature space is another important factor. In Figure 26-(b), it can be seen that the proposed technique becomes more and more effective as the size of the region increases. For example, when the area is set to 0.2σ and the amount of the noise is 20%, the proposed framework is unable to show any advantage over the baseline system. However, tested with the same amount of the noise but a larger size, say σ, it is seen that the error rate is reduced by 17.5% relatively.

In Figure 27, we also present performance comparison curves for the cases when the number of labeled sample is 50. As shown in Figure 27, the systems with proposed technique (the dotted lines) still outperform the baseline system (the solid line), although the gap between the solid line and the dotted lines is narrowed. In other words, when we have a large set of labeled samples, the use of an agreement function might not be significantly advantageous. In particular, the maximum relative error rate reduction is 11.4% when 30% of the training samples have been corrupted by the disagreement noise as shown in Figure 27-(a). When the amount of disagreement noise is 20%, the relative error rate reduction is decreased to 7.4%. However, this may be because the performance of the baseline system has already been reached to the Bayes optimal error rate when enough number of labeled samples are available. In particular, the Bayes error rate when the amount of disagreement



(a) Performance comparison curves on the unlabeled set. The black solid line corresponds to the baseline system, while dotted lines represent the error rates of the proposed multi-view SSL framework.



(b) Performance comparison curves on the test set. The black solid line corresponds to the baseline system, while dotted lines represent the error rates of the proposed multi-view SSL framework.

Figure 27: Performance comparison graphs between the baseline system and the proposed multi-view SSL technique when the number of labeled samples is set to 50. The solid line corresponds to the baseline system that does not use the agreement function. The dotted lines are for the proposed framework. x-axis represents the amount of disagreement noise, while y-axis is an error rate. For both (a) and (b), as the amount of disagreement noise and the area in feature spaces where the noise can occur increase, the proposed framework becomes more and more effective, while it is not as effective as in the case when the number of labeled samples is 20. Interestingly, the proposed framework appears to work better for test data than for unlabeled data samples.

noise is 20% is 12.5%, which is the exact error rate for the proposed technique as seen in Figure 26-(b). Nevertheless, this also implies an agreement function is typically desirable for multi-view SSL because the size of a labeled set is rarely big enough.

5.4 Multi-view SSL on the TREC05 spam corpus

Recently, text-based learning filters have grown in sophistication and effectiveness in filtering email spam [23, 30, 82]. However, in response, spammers have adopted a number of countermeasures to circumvent these text-based filters. Currently, one of the most popular spam construction techniques involves embedding text messages into images and sending either pure image-based spam or a combination of images that contains spam messages and text messages that are typically seemingly legitimate. This strategy, usually called *image spam* has been successful in bypassing text-based spam filters, posing a new challenge for spam researchers [107]. Figure 28 illustrates some examples of such images extracted from image spam emails.

Figure 28: Examples of images containing spam messages

Attempts to use optical character recognition (OCR) techniques to convert spam images back to text for processing by text-based filters have been foiled [62]. An effective response by spammers is the application of CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) techniques, which are designed to preserve readability by humans but capable of effectively confusing the OCR algorithms. Thus, in [20], four key properties of spam images, such as color moments, color heterogeneity, conspicuousness, and self-similarity were used to train a discriminative classifier with multi-class characterization to detect spam images. Experimental results on the TREC05 spam corpus achieved 86.6% detection rate with 19.1% false alarm rate. Although these results imply that spam image detection is promising artillery for a battle against *image spam*, it is also demonstrated that

the sole use of image-based filtering techniques would not be enough because of the high false alarm error rate. As a response, [19] leveraged text-based spam filters, which perform extremely well in identifying legitimate emails, in addition to the image-based spam filters. Note that, however, it studied a combination of image-based filters and text-based filters in a supervised setting, while SSL techniques for spam filtering remained relatively unexplored.

As mentioned in the beginning of this chapter, one of the properties of *image spam* is the discrepancy between the contents of images and those of texts. Figure 29 illustrates such an example where texts with legitimate messages and images with spam messages are extracted from a single email. We can also see that in Figure 30, the words used in a spam

Only a superficial soul fears to fraternize with itself. The ultimate measure of a man is not where he stands in moments of comfort and convenience, but where he stands in times of challenge and controversy.

Beer is proof that God loves us and wants us to be happy. Memory is a child walking along a seashore. You never can tell what small pebble it will pick up and store away among its treasured things.

Compassion is the antitoxin of the soul; where there is compassion even the most poisonous impulses remain relatively harmless. The best proof of love is trust. Love takes off masks that we fear we cannot live without and know we cannot live within.

(a) A textual part of an image spam email

(b) An attached image of an image spam email

Figure 29: An example of an image spam email. Note the text content does not contain any spam messages while the image advertises illegal medication

email (e.g., Figure 30-(a)) are very similar to those used in a legitimate email (e.g., Figure 30-(b)). According to the definition of the disagreement noise presented in the beginning of Chapter 5 (i.e. given multiple features, the disagreement noise occurs if more than two class labels are assigned to a single data sample depending on the features to which a labeler refers), this characteristic of image spam, shown in Figures 29 and 30, can be considered as a typical example of the disagreement noise; no one will label texts in Figures 29-(a) and 30-(a) as spam if no other information (e.g., the address of a sender or the attached images, etc) is given. Thus, we can expect that the use of the proposed multi-view SSL framework to image spam will be particularly beneficial.

Gissin said rescue operations were continuing Sunday night. The attack "indicates that unless there is decisive and sustained effort taken to dismantle the terrorist organization, it will be impossible to move towards normalizations and towards political negotiations," Gissin told a news crew. "And I think the responsibility on that lies with the Palestinian Authority." Shortly after the first blast, a second explosion was heard in southern Gaza, but its precise location was not immediately known. Hamas, in a phone call to CNN, said it had set off the first explosion near Rafah in cooperation with a group called the Fatah Hawks.

There was no immediate information available on that group, although it was believed to be linked to the Fatah movement formerly led by the late Palestinian leader Yasser Arafat.

(a) A text message extracted from an image spam email

An Irishman named O'Malley went to his doctor after a long illness. The doctor, after a lengthy examination, sighed and looked O'Malley in the eye and said, "I've some bad news for you. You have cancer, and it can't be cured, you'd best put your affairs in order."

O'Malley was shocked and saddened; but of solid character, he managed to compose himself and walk from the doctor's office into the waiting room. To his son who had been waiting, O'Malley said, "Well son. We Irish celebrate when things are good, and we celebrate when things don't go so well. In this case, things aren't so well. I have cancer. Let's head for the pub and have a few pints."

(b) A text message extracted from an legitimate email

Figure 30: A comparison of a text message from an image spam email with that from legitimate email. Note how the text message from a spam email appears to be legitimate compared to that from a legitimate email

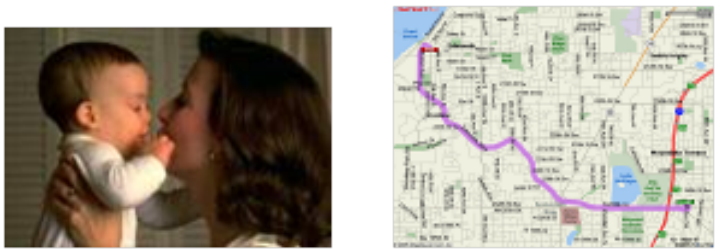
5.4.1 Feature extraction techniques

In this section, we briefly explain the image feature extraction methods for image spam for self completeness, which was originally proposed in [20], in Section 5.4.1.1. Please refer to [20] for more details. We also discuss the feature extraction techniques for the text component of image spam in Section 5.4.1.2.

5.4.1.1 Image features

The first distinctive property of spam images is color moments. In spam images, several notable color characteristics can be found such as discontinuous distributions, high intensity, dominant peaks, etc. The simplest way to extract these color characteristics is to use color histograms [95]. However, since most of the information is embedded in low-order moments, color moments can be used instead [94]. In our experiments, the first and second central moments were computed through the following simple steps: (a) all images were transformed to the HSV color space [45], and (b) in the HSV color space, the first and second central moments were computed for every channel.

The second property of spam images is color heterogeneity. Typically, legitimate images



(a) (b)

Figure 31: Examples of images extracted from legitimate emails

convey a much larger number of colors than spam images. For example, in a person's face, large variations can be observed, which account for shades or illuminations in the face. In contrast with legitimate images, color in spam images usually stays constant. The background is generally filled with the same colors, and a spam message embedded into the image usually consists of a handful of colors. To see this, please compare the legitimate image in Figure 31-(a) and the spam images in Figures 28-(a) to -(d). In this work, this property, referred to as color heterogeneity, is extracted as follows: first quantizing an image with at most N colors through a minimum variance quantization algorithm and then measuring the root-mean-squared error between the original image and the quantized image.

The third property of spam images is called color conspicuousness. Spammers want spam messages to be easily noticeable to recipients so that desired actions can be incurred (e.g., reading the message, clicking a link, etc.). Thus, it is natural for spam images to use highly contrasted colors. In practice, we can indeed observe that many spam images use highly saturated colors with contrasting white or black or white contrasting black. This time, the comparison between Figure 31-(b) and Figures 28-(a) to -(d) clearly reveals this property of spam images. In this work, this feature is extracted from an SV plane, which is a subspace of an HSV color space. In particular, we first learn M centroids of the pixel values of an image in the SV plane. Given the M centroids, we compute an average of the shortest distance between the centroids and three points (0,0), (0,1), and (1,1).

As for the fourth property, we extracted self-similarity of an image. In spam images, some characteristic patterns are usually repeated. Specifically, in spam images, it is more likely to see a uniform background than from legitimate images. Moreover, text in spam

images usually exhibit similar fonts and sizes. To extract the self-similarity, we need to learn representative patterns from data first. To this end, we first segment a whole image into several macro-blocks, where the size of a macro-block is globally set to 32x32. We then compute a log-Gabor filter bank [60] from each block and concatenate the means and the variances of the outputs of the filter bank. Next, we perform clustering, where the centroid of each cluster is considered as the representative patterns. Once such patterns are identified, we index the macro-blocks in images back using the learned pattern (i.e., to associate each block with the closest centroid) with which the histogram of indices becomes the extracted self-similarity vector.

Finally, we also find that meta-data of images are also useful to distinguish spam images from legitimate images. For instance, the ratio of a file size to the dimension of an image (i.e., the compression ratio) is usually smaller for spam images than legitimate images. This is because spam images have less foreground information. Entropy evaluated from the file name of an image is also a good indication of spam images because the name of spam images is usually generated randomly by a machine. Lastly, compression algorithms (e.g., JPEG, GIF, PNG, etc.) provide useful information for spam image identification.

5.4.1.2 Text Feature Extraction

To extract text features, we parsed emails in the corpus by removing email headers and multipurpose internet mail extensions (MIME) tags. For emails embedded into hypertext markup language (HTML) tags, we also parsed such tags so that only pure text components were remained. Then, term-frequency-inverse-document-frequency (*tf-idf*) normalization was carried out. This *tf-idf* normalization is a commonly used technique in text categorization. The *tf* score indicates the relevance of a word to a document, which is usually computed as the frequency of the word with in the document. On the other hand, the *idf* score measures the indexing power of a certain term evaluated by the inverse of the number of documents in which the term appears. More formally, suppose that there exists a collection of emails $\mathcal{D} = \{d_1, \ldots, d_N\}$ and a lexicon $\mathcal{T} = \{t_1, \ldots, t_M\}$. Then, using the *tf-idf* normalization, we can represent the i^{th} email d_i with a M-dimensional vector v_i whose

element v_{ij} was given by

$$v_{ij} = (\frac{a_{ij}}{\sum_k a_{kj}}) \times \log(\frac{N}{|d_j : t_i \in d_j|}), \tag{91}$$

where a_{ij} is the number of occurrences of the i^{th} term in the j^{th} email and $|d_j : t_i \in d_j|$ is the number of documents in $\mathcal{D}$ in which the i^{th} term appears.

5.4.2 Experimental results

To test the proposed multi-view SSL framework, out of 92189 emails in the TREC05 spam corpus (52790 emails for spam / 39399 emails for legitimate), we filtered out emails that did not contain any images. Because spammers had copied the same emails over and over to minimize the cost of generating a new type of spam, the remaining image spam had many duplicates. After eliminating such duplicates, we had 1377 image spam emails left in which there were 1170 spam emails and 207 legitimate emails. Given these 1377 emails, we performed 80:20 splits (80% of them for training and the remaining 20% for testing) 100 times to compute an average classification error rate. To create a semi-supervised environment, we further divided the emails reserved for training (1102 out of 1137), selecting 5%, 10%, and 20% of them. We then grouped the chosen samples as labeled sets and the rest of the training samples were considered as unlabeled data samples.

Features were extracted based on the feature extraction techniques discussed in Section 5.4.1. As a result, the image feature generated a 86-dimensional vector (a 12-dimensional vector for color moments, two 1-dimensional vectors for color heterogeneity and color conspicuousness, a 64-dimensional vector for self-similarity, and a 8-dimensional vector for meta-data) and the text feature produced a 17896-dimensional vector. To create an unified RKHS, we also computed a kernel matrix for each of them. In particular, an RBF kernel with a bandwidth h set to 1 was used for the image feature where the associated distance function was the Euclidean distance. For the text feature, an RBF kernel with its bandwidth h set to 1 was again computed in which a cosine distance function was chosen for the distance measure of the kernel.

As for the parameters λ_1, λ_2 in Eq. (89), the best performing values were chosen within the range of $1e^{-6}$ to 1, incremented by a factor of $1e^2$. On the other hand, another parameter

μ in Eq. (89) was set to 1 throughout experiments. In the proposed technique, parameters for an agreement function Ξ, namely the number of nearest neighbors q and the size of the bandwidth of an RBF kernel h were configured as follows: $q = 5$ for both image and textual features, and h was set to an average Euclidean distance (or cosine distance) among feature vectors for image feature (or for text feature). Similar to the experimental setups in the artificial data sets, the RLSR technique was adopted for our classifier. Then, we prepared for the following four systems where the first three were the baseline systems and the last one was the proposed technique:

- an RLSR classifier with the image feature only (image-only)
- an RLSR classifier with the text feature only (text-only)
- an RLSR classifier with the co-regularized multi-view SSL technique without the agreement function (w/o agreement)
- an RLSR classifier with the proposed multi-view SSL technique using the agreement function (w/ agreement)

In Table 4, we report the classification error rates for the proposed multi-view learning algorithm (i.e., w/agreement) compared them with single feature cases (i.e., image-only and text-only) while varying the size of labeled data samples. As noted, the first three rows correspond to the error rates for unlabeled data sets and the next three rows are for the test data sets. In Table 4, it is clearly seen that by using multiple features rather than a single feature, classification error rates are reduced significantly for all cases. In particular, using the image feature only, 9.71% of a classification error rate was achieved for unlabeled samples when 10% of training data were used as a labeled set. On the other hand, with the same number of labeled samples, the sole use of the text feature produced 7.56% of a classification error rate. However, when the image feature and the textual feature were combined, only 5.8% of unlabeled samples were misclassified. As for test data, the proposed framework also outperforms single-feature systems significantly for all various sizes of labeled data sets. Therefore, it can be said that our multi-view SSL framework is able to effectively

combine multiple features, producing significant performance gains over the cases when a single feature is used separately.

		5%	10%	20%
Unlabeled	image-only	9.57	9.76	8.62
	text-only	9.95	7.56	4.45
	w/ agreement	**7.52**	**5.8**	**4.13**
Test	image-only	10.02	9.71	8.71
	text-only	10.12	6.81	4.52
	w/ agreement	**7.73**	**5.87**	**4.18**

Table 4: Comparisons of classification error rates on the TREC05 spam corpus between single-view cases (image-only and text-only) and a multi-view case (w/ agreement). The top three rows show the classification error rates of the three systems on unlabeled data sets averaged over the 100 runs. The bottom three rows correspond to the classification error rates of the chosen three systems on test data averaged over the 100 runs. Bold font indicates statistically significant cases. It can be seen that for all cases, the use of multiple features through our proposed framework outperforms systems using a single feature separately.

Next, we compare two co-regularized SSL algorithms in Table 5, one of which exploits an agreement function and the other does not (i.e., w/ agreement vs. w/o agreement), to highlight the effectiveness of the agreement function when the disagreement noise presents as we increase the size of a labeled data set from 5% to 20%. In Table 5, the top three

		5%	10%	20%
Unlabeled	w/o agreement (%)	8.06	6.29	4.44
	w/ agreement (%)	**7.52**	**5.8**	**4.13**
	p-value	1.81×10^{-5}	8.87×10^{-8}	7.06×10^{-6}
Test	w/o agreement (%)	8.14	6.27	4.36
	w/ agreement (%)	**7.73**	**5.87**	4.18
	p-value	5.18×10^{-4}	6.69×10^{-6}	0.05

Table 5: Comparisons of classification error rates on the TREC05 spam corpus between a baseline system (w/o agreement) and our proposed multi-view SSL framework (w/ agreement). The top three rows show the classification error rates of the baseline and our system on the unlabeled data sets averaged over 100 runs followed by the corresponding p-value evaluated with paired two-tail t-tests. The bottom three rows present average performances on the test data sets over 100 runs and the corresponding p-values. Bold font means statistically significant cases.

rows show the classification error rates of the baseline system (w/o agreement), those of the proposed technique (w/ agreement), and the corresponding p-values evaluated by paired two-tail t-tests for the unlabeled data sets, respectively. The bottom three rows present similar

quantities except that they are measured with the test data sets. Comparing the two systems, it can be seen that the proposed multi-view SSL framework that uses the agreement function outperforms the baseline system. Specifically, for unlabeled data, relative classification error reductions are 6.7%, 7.8%, and 7.0% when the sizes of labeled samples are 5%, 10%, and 20% of the entire training data, respectively. Similarly, the classification error rates are reduced by 5%, 6.4%, and 4.1% relatively when the agreement function is exploited for test data in the same configurations. When we evaluate p-values using paired two-tailed t-tests, they indicate that the performance improvements of the proposed technique over the baseline system (w/o agreement) are statistically significant for unlabeled data sets. For test samples, on the other hand, the systems with an agreement function outperform the baseline systems statistically significantly when the size of labeled samples are 5% and 10%, but not when the size of a labeled set is 20% of the training data samples. This can be understood by the fact that given enough number of labeled samples, the baseline system (w/o agreement) might already reach a similar performance of a fully-supervised system. It is also interesting to see that the amount of the effectiveness of the proposed framework is larger on unlabeled samples than on test data, which presents a potential benefit of the proposed system in a transductive setting, where test data are used during a training phase as unlabeled samples.

5.5 Summary

In this chapter, we proposed an algorithm to learn an agreement function for a multi-view SSL framework with which the amount of agreement we wanted to impose on unlabeled data could be quantified. The key findings of this chapter can be summarized as follows: (a) the value of the agreement function can be modeled as the amount of local information that is shared among different views, and (b) the use of the agreement function in a presence of the disagreement noise is advantageous in reducing classification errors. Therefore, if two nearby feature vectors in one feature space are located in different clusters in the other feature space, one should relax the agreement assumption. On the other hand, if two feature vectors are close to each other in both feature spaces, the agreement assumption should remain enforced.

From a generalization error minimization perspective, by allocating the assumption in such a data-driven way, we expect the true model more likely to be included into a hypothesis space while keeping the complexity of the space in a similar level. Comparing the system that applied the equal amount of the agreement assumption to every unlabeled sample with the system that used the agreement function on artificial data sets that contained the disagreement noise, the classification error rates were reduced when the agreement function was used.

We also investigated an application of the proposed learning framework to a real-world machine learning problem, image spam detection. Image spam is a typical example where multiple views (e.g., text messages and attached images) with the disagreement noise exists. This is because spammers usually embed spam messages into images and camouflage such spam messages with legitimate text. Compared with the cases (a) when a single feature was used individually and (b) when no agreement function was applied, our experimental results demonstrated that the use of an agreement function was indeed beneficial, especially to unlabeled samples in terms of improving prediction accuracy. In fact, this property is advantageous to semi-supervised incremental learning because of the fact that semi-supervised incremental learning mainly concerns with how to correctly create the label information for unlabeled samples. In Chapter 6, therefore, we will discuss a simple but effective method to integrate the multi-view learning technique presented in this chapter with the semi-supervised incremental learning technique discussed in Chapter 4.

Chapter VI

DISCRIMINATIVE SEMI-SUPERVISED INCREMENTAL LEARNING APPROACH WITH A MULTI-VIEW PERSPECTIVE

One of the fundamental issues in semi-supervised incremental learning is how to ensure convergence of an incremental learning process while getting away from sub-optimal solutions. In many incremental learning algorithms, the corresponding learning procedures either converge into sub-optimal solutions or diverge from their initial models. Confidence score based algorithms are typical examples of semi-supervised learning techniques that sub-optimal solutions are often obtained. As discussed in Chapter 2, the Co-training has been proved to be a viable solution to this sub-optimal-solution problem. However, the underlying assumption in the Co-training, outputs from two distinct classifiers are conditionally independent given class labels, has made the applicability of the Co-training somewhat restrictive. On the other hand, the performance measure based technique proposed in [112] has also attempted to tackle the sub-optimality by evaluating a certain performance measure directly, but such evaluation usually takes too many computational resources so that it also might not be an effective solution to the issue.

One easy yet more principled approach to such an issue is to create better performing initial models. In fact, the experimental results in Chapter 4 have demonstrated the importance of having good initial models. It has been seen that while attempting to address the sub-optimality issue, the learning procedures diverge if the initial models do not perform sufficiently well. Moreover, more powerful initial models generate less labeling errors when incorporating unlabeled samples into the learning process. Thus, the risk of classification models divergence will also be reduced. However, it should be noted that in Chapter 4, the main recipe for improving the performance of initial models was to exploit a larger set of labeled data samples, which is in many cases, unrealistic. Typically, we do not have enough number of labeled samples to train good initial models. In contrast, in this chapter, we

address this issue by starting with better performing models as follows:

- We unify multiple features through the multi-view learning technique discussed in Chapter 5 because the more features we have, the more likely that a model performs better.

- We learn the initial models using discriminative learning approaches, such as a kernelized MFoM learning approach introduced in Chapter 3, because discriminative learning tends to be more robust in a *small sample size* situation than generative learning approaches such as Gaussian mixture models.

In this chapter, we also investigate the feasibility of a cohesive semi-supervised incremental learning system that integrates the multi-view learning technique proposed in Chapter 5 with the semi-supervised incremental learning approach proposed in Chapter 4, namely a discriminative semi-supervised incremental learning approach with a multi-view perspective. We highlight the effectiveness of the integrated system by experimenting it with the TREC05 spam corpus. Experimental results show that the use of the multi-view learning technique along with the semi-supervised incremental learning approach indeed improves both the robustness and the performance of the incremental learning technique significantly. The rest of this chapter is organized as follows. In Section 6.1, we present our integrated system in more detail. Then, experimental results are given in Section 6.2 followed by a summary with concluding remarks in Section 6.3.

6.1 System overview

A block diagram of an integrated discriminative semi-supervised incremental learning system is presented in Figure 32. As can be seen, the integrated system inherently consists of two main building blocks: (a) the component to handle multiple features in the front-end and (b) the semi-supervised incremental learning system at the back-end. Typical procedures of the system can be explained in more detail as follows: suppose that a certain data set is given (e.g., the TREC05 spam corpus). Then, several feature extraction modules are applied to the data set, generating multiple feature vectors. Note that at this stage some

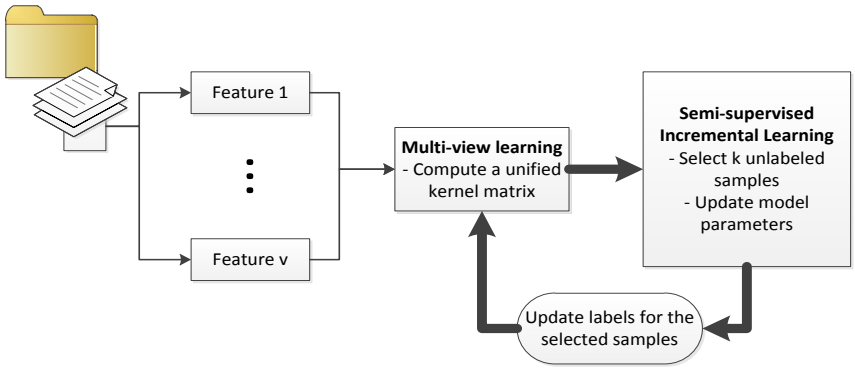


Figure 32: A block diagram of the proposed discriminative semi-supervised incremental learning approach with a multi-view perspective. Given a data set, a multi-view handler located in the middle collects features and generates a unified kernel matrix. Then, using this kernel matrix, a semi-supervised incremental learning technique recommends unlabeled samples for parameter update. While updating parameters, the label information of the selected unlabeled samples is also generated. This newly generated class labels are then exploited to recompute the kernel matrix, which will be subsequently used to retrain classification models. We use thin arrows to differentiate the steps that are done only once from the constant work flows for incremantal learning, marked with thick arrows.

feature vectors can be grouped together to form a more meaningful feature vector (e.g., putting image-related features together to form a single image feature vector). Next, the resulting feature vectors are fed into the multi-view learning module, generating a unified kernel matrix. Given this unified kernel matrix, an ensemble of classifiers is trained by applying various learning algorithms. The learned classifiers are then used to compute expected error reduction measures, confidence scores, class prior distributions, and Zipf distributions, all of which are exploited to recommend unlabeled samples that will be incorporated into the incremental learning process. Finally, the multi-view learning module creates an updated kernel matrix using the selected unlabeled samples along with their predicted class labels. This matrix is then applied back to the semi-supervised incremental learning module, completing a closed loop. This closed loop is highlighted in Figure 32 with thick arrows.

Note that one might pursue different configurations from Figure 32. For example, instead of using a multi-feature handler that transforms multiple features into a unified kernel matrix located at the middle of the system block diagram as shown in Figure 32, one can create a

collection of classifiers, each of which is trained on a single feature each. Then, consider this set of classifiers as the ensemble of classifiers with which the selection score defined in Eq. (66) is evaluated. While this approach seems viable at first glance, there exists one issue that needs to be addressed. In particular, because classifiers are learned based on different features, they might exhibit a wide spectrum of performances, so that the confidence level of the estimated parameters should be taken into account when unlabeled samples are selected for training. However, doing so with only a small amount of samples might be difficult and additional noise can be introduced. In contrast, the configuration illustrated in Figure 32 is straightforward and relatively easy to implement.

We conclude this section by summarizing the detailed algorithmic steps of the proposed discriminative semi-supervised incremental learning approach with a multi-view perspective in Algorithm 4. It should be noted that the differences between Algorithm 3 and Algorithm 4 are steps to update parameter vectors after the multi-feature handler (see Figure 32).

6.2 Experimental results

To test our integrated system, we used the TREC05 spam corpus, a image spam data set introduced in Chapter 5. Since one could divide the feature spaces into two fairly easily (i.e., image and text), this data set was a very good test case for the proposed framework. Similar to our discussion in Chapter 5, out of 92189 emails, we extracted 1377 emails that contained both the image and textual components after eliminating duplicated samples, resulting in a total of 207 legitimate emails and 1170 spam emails. We then created four different kernel matrices, namely, *image-only*, *text-only*, *w/o agreement function*, and *w/ agreement function*, for comparisons as follows:

- an RBF kernel with image feature only using the Euclidean distance and the bandwidth set to 1 (image-only).
- an RBF kernel with text feature only using the cosine distance and the bandwidth set to 1 (text-only).

Algorithm 4 A semi-supervised incremental learning with a multi-view perspective

prepare $\mathcal{U}^0$ and $\mathcal{L}^0$
$K^0 \leftarrow k(x,z)$ for $x,z \in \mathcal{U}^0$ and $\mathcal{L}^0$ according to Eq. (89)
initialize $\boldsymbol{\theta}^0$ with K^0
$t \leftarrow 0$
$C \leftarrow$ the number of classes
repeat
  compute Eq. (67), Eq. (61) and Eq. (66) for all samples in $\mathcal{U}^t$
  estimate the class prior distribution using all samples in $\mathcal{U}^t$ based on Eq. (68)
  $k_u^t \leftarrow$ the number of samples to be selected at time t
  $\mathcal{U}_y^t \leftarrow \{x \,|x \in \mathcal{U}^t,\ \delta(x;\boldsymbol{\theta}^t) = y,\ s(\delta(x;\boldsymbol{\theta}^t)) \geq \tau\}$ for $y = 1,\ldots,C$
  $N_{u_y^t} \leftarrow |\mathcal{U}_y^t|$ for $y = 1,\ldots,C$
  $\tilde{\mathcal{U}}_y^t \leftarrow \{x_{(i)} \,|x_{(i)} \in \mathcal{U}_y^t,\ s(\delta(x_{(1)},\boldsymbol{\theta}^t)) \geq s(\delta(x_{(2)},\boldsymbol{\theta}^t)) \geq \cdots \geq s(\delta(x_{(N_{u_y^t})},\boldsymbol{\theta}^t))\}$
  compute a Zipf distribution based on Eq. (69) for all $\tilde{\mathcal{U}}_y^t, y = 1,\ldots,C$
  while $|\mathcal{S}^t| < k_u^t$ **do**
    pick a candidate class m according to the estimated class prior distribution
    if $|\tilde{\mathcal{U}}_m^t| \neq 0$ **then**
      pick x according to the computed Zipf distribution for the class m
      $\tilde{\mathcal{U}}_m^t \leftarrow \tilde{\mathcal{U}}_m^t \backslash x$
    else
      continue
    end if
  end while
  $\mathcal{L}^{t+1} \leftarrow \mathcal{L}^t \cup \mathcal{S}^t$
  $\mathcal{U}^{t+1} \leftarrow \mathcal{U}^t \backslash \mathcal{S}^t$
  $K^{t+1} \leftarrow k(x,z)$ for $x,z \in \mathcal{U}^{t+1}$ and $\mathcal{L}^{t+1}$ according to Eq. (89)
  update $\boldsymbol{\theta}^{t+1}$ with K^{t+1}
  if $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t$ **then**
    break
  else
    $t \leftarrow t+1$
  end if
until $|\mathcal{U}^t| = 0$

- an unified kernel with both the RBF kernel from image feature and the RBF kernel from text feature combined. NOT using an agreement function (w/o agreement function).

- an unified kernel with both the RBF kernel from image feature and the RBF kernel from text feature combined. Using an agreement function (w/ agreement function).

Note the similarity of the kernel matrices prepared here and those used in Chapter 5, where the effectiveness of the multi-view SSL framework was demonstrated. This was because

our main concern for this set of experiments was to see the potential advantages of tying the multi-view learning framework together with the semi-supervised incremental learning technique proposed in Chapter 4.

The parameters needed for the cases of *w/o agreement function* and *w/ agreement function* were borrowed from Chapter 5, where the parameters were found based on the classification error rates on the validation sets using a Regularized Least Square Regression (RLSR) technique. The parameters for the semi-supervised incremental learning were picked using the protocol described in Chapter 4. Specifically, we chose the convex combination parameter γ and the threshold τ based on the performances on the test data sets during the first few iterations, each of which was incremented by 0.02 within a range of 0.44 to 0.54 for γ and from 0.86 to 0.94 for τ. On the other hand, a positive constant η for a Zipf distribution and the size of the selection set $\mathcal{S}^t$ were set to 3.6 and 3% of the number of training examples available at time t, respectively, for simplicity. The sets of labeled, unlabeled, and test data samples were created by using the exact same splits generated in Chapter 5. This way, we could infer how well the proposed semi-supervised incremental learning framework performed, compared to a semi-supervised learning technique trained in a batch mode. The size of an ensemble for semi-supervised incremental learning was chosen to be four and we used randomized kernelized MFoM learning techniques discussed in Chapter 3 for the classifiers in the ensemble. In particular, 200 samples were randomly chosen out of the set of 1102 samples with which a subspace of a function space was built. As for the size of initially labeled samples, we experimented with two different set sizes, say 5% and 10% of the training set.

First, we have drawn performance comparison curves for the above-mentioned four systems when the number of an initial labeled set is 5% of the training examples in Figure 33. From the curves in Figure 33, it can be clearly seen that the initial models are well converged for all cases, showing the robustness of the semi-supervised incremental learning framework discussed in Chapter 4. More importantly, the proposed learning framework (i.e., *w/ agreement function*, which corresponds to the bottom curve in Figure 33) achieves the best performance among the four different tested systems. In particular, comparisons of the

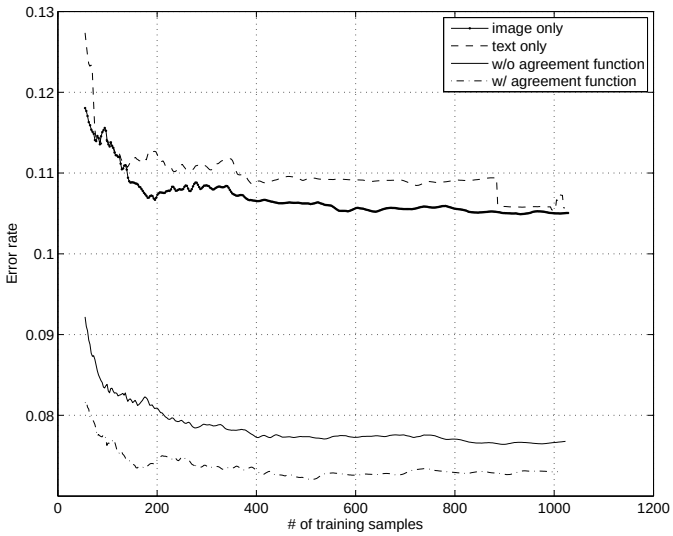


Figure 33: Performance comparison graphs of the proposed semi-supervised incremental learning framework on the TREC05 spam corpus when the size of initially labeled set is 5% of the training set. Note that the proposed framework (the bottom curve) outperforms all other cases. It can be also clearly seen that the use of multi-view learning techniques improves the classification error rates significantly as well the stability of the curves (see the top two curves compare them with the bottom two curves.). x-axis represents the number of training samples collected at a certain time t, while y-axis represents error rates evaluated over test sets.

proposed technique with systems using a single feature (i.e., *image-only* and *text-only*, which correspond to the top two curves in Figure 33, respectively) demonstrate that the proposed technique experiences a relative error rate reduction of 30.5% in terms of final classification. Moreover, comparing the proposed system with the performance of a multi-view approach without an agreement function (i.e., *w/o agreement function*, the second curve from the bottom in Figure 33) highlights the effectiveness of the use of an agreement function; the classification error rate is further reduced by 4.6% relatively. Interestingly, it can be seen that the proposed framework achieves even better classification performance (i.e., the error rate of 7.3%) than the error rate of 7.73% reported in Chapter 5, showing the benefit of using discriminative classifiers, such as kernelized MFoM learning approaches.

In Figure 34, we show similar performance comparison curves except that the size of an initial labeled set is now increased to 10%. Again, similar tendencies to those in Figure 33 are observed. In particular, although the improvement from incorporating an agreement

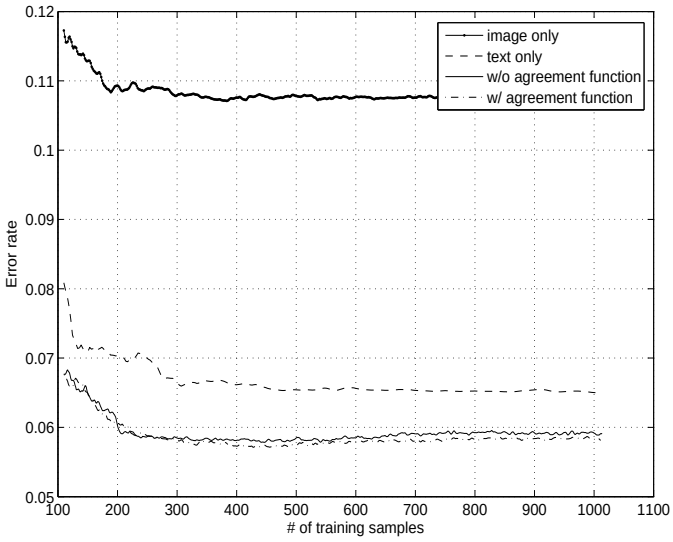


Figure 34: Performance comparison graphs of the proposed semi-supervised incremental learning framework on the TREC05 spam corpus when the size of initially labeled set is 10% of the training set. Refer to the descriptions in Figure 33 for more details. The proposed framework still performs the best in terms of the final performances, showing a clear advantage of using a multi-view handler as well as the agreement function.

function is now marginal, one can easily see multi-feature cases (*w/o agreement function* and *w/ agreement function*) outperform the single-feature cases (*image-only* and *text-only*) all the time. More precisely, compared to the image only case (i.e., the top curve in Figure 33), the proposed learning framework exhibits a relative error rate reduction of 45.8% from 10.7% to 5.8%. On the other hand, 10.8% of relative performance improvement is observed when comparing the proposed technique with the text-only case (i.e., the second curve vs. the bottom curve in Figure 33). Interestingly, there seems to be no observed improvement when only an image feature is used even if the size of an initial labeled set is increased. This, in a way, illustrates the possibility that a learning process becomes unstable when only a single feature is used in semi-supervised incremental learning.

6.3 Summary

In this chapter, we investigated a possibility to further improve the semi-supervised incremental learning framework discussed in Chapter 4 by integrating it with the multi-view learning approach proposed in Chapter 5. Because a kernel function formulated in Chapter

5 was able to combine multiple features, such integration was done by applying the kernel function directly into the unlabeled sample selection system for incremental learning. The end result was a novel semi-supervised incremental learning framework with a multi-view perspective. Experimental results on the image spam extracted from the TREC05 spam corpus demonstrated that the use of a multi-feature handler during incremental learning procedures indeed reduced classification error rates by 30% on average, as compared to when a single feature was used. In sum, the integrated semi-supervised learning framework enables us not only to take advantage of better initial models for robustness, but also to produce accurate image concept characterization systems.

Chapter VII

CONCLUSION AND FUTURE WORK

This dissertation discussed the development of a novel semi-supervised incremental learning framework with a multi-view perspective on the ground of many machine learning techniques, such as discriminative learning, semi-supervised incremental learning, multi-view semi-supervised learning, and so on, and demonstrated the effectiveness of the developed framework with an application to various image concept modeling problems such as handwritten digit recognition, object recognition, and image spam detection problems. Recent advances of image acquisition techniques make semi-supervised incremental learning particularly attractive to image concept modeling. Ideal semi-supervised incremental learning should make use of all possible information extracted from data in addition to all prior knowledge accrued from past experiences. However, because of the small number of labeled data due to limited resources, collecting useful information for semi-supervised incremental learning is often difficult. Moreover, even after some useful pieces of information are gathered, it is usually not clear how to tie each piece together in a principled way. Thankfully, in the literature, quite a bit of related research had been conducted, which provided us with ample hints for what information should be used to where and how, as discussed in Chapter 2.

In Chapter 3, a discriminative learning algorithm, namely a kernelized maximal-figure-of-merit (kMFoM) learning approach, was investigated. The qualitative analysis on some image data sets in Chapter 3 indicated that non-linearization of discriminant functions would have advantages in characterizing image concepts. We showed that by preserving the property of the original MFoM learning approach (i.e., a variety of performance metrics are directly optimized during the learning process), the proposed kMFoM technique was capable of modeling various image concepts effectively. Because non-linearization through kernel functions entailed higher computational complexity, we exploited a subspace distance minimization

technique in which a subset of training data samples were chosen to be trained with. To retain the coverage of a function space constructed with the subset, a subspace distance minimization technique using the Nystöm extention was developed. A set of experiments comparing the learning time with the performance showed that the performance of the proposed framework was comparable to that of a system trained with the entire training data, highlighting the effectiveness of the proposed system.

On the other hand, in Chapter 4, we answered the fundamental question of semi-supervised incremental learning, how to arrange a sequential use of unlabeled samples, by proposing a novel semi-supervised incremental learning framework in which an expected error reduction was computed based on a Bayesian decision theory. Unlike the confidence scores that were used in the past, the use of the expected error reduction intended to directly measure contributions of unlabeled samples to reducing classification errors so that we could avoid a potential sub-optimality problem of the confidence score based methods. In sum, the proposed framework incorporated *informative* and *dependable* unlabeled samples more quickly by blending an expected error reduction measure and a confidence score. One unique aspect of the proposed learning framework was that it enabled us to put various information together to make the framework robust. In particular, the virtue of an ensemble of discriminative classifiers was taken advantage of for reliable estimation of the expected error reduction. Class prior distributions were also used to prevent the *class imbalance problem* by nominating a candidate class first and then choosing unlabeled samples among the samples predicted to be in that class. Experimental results on various image concept modeling problems clearly showed the effectiveness of the proposed framework with a couple of remarks. First, within the ensemble, it was advisable to have classifiers that produced diverse classification outputs. Second, the initial models should be sufficiently good to reduce a potential bias and the risk of including incorrect class labels.

As a viable approach to having good initially trained models, in Chapter 5, we investigated a method to deal with multiple features in a semi-supervised setting. Multiple features play a crucial role in image concept modeling because images are inherently perceived through multiple channels. In the literature, early fusion has been a predominant

technique for integrating multiple features in semi-supervised cases because late fusion approaches tend to over-fit to labeled samples. Conventionally, in the early fusion approaches, it has been assumed that prediction results of unlabeled samples obtained from individual features should be the same. In Chapter 5, we argued that such an assumption should be imposed based on the amount of the overlap of local structures along different features. We then proposed an agreement function to convey such information. Experimental results on artificially generated data sets showed that the use of the agreement function was indeed useful especially when disagreement noise presented. We also demonstrated the effectiveness of the proposed multi-view learning framework on an image spam detection problem.

Given the techniques developed in Chapters 3, 4, and 5, an interesting observation was made in Chapter 6; the fusion technique in Chapter 5 provided us a natural way to combine the technique with the semi-supervised incremental learning framework presented in Chapter 4 and the kMFoM learning technique discussed in Chapter 3. Therefore, in Chapter 6, we further proposed an integrated semi-supervised incremental learning framework, namely a discriminative semi-supervised incremental learning framework with a multi-view perspective. Specifically, we formulated a closed-form solution for a unified kernel function that merged different feature spaces altogether. Next, we used the kernel function to train kMFoM classifiers from which the quantities needed to recommend unlabeled samples were calculated. The experimental results on one of the image concept modeling problems, image spam detection, showed a clear advantage of the integrated system in that we were able to start the incremental learning process from good-performing initial models by taking advantage of several complementary features, which in turn improved the stability of the overall learning procedures.

7.1 Contributions of this dissertation

The contributions of this dissertation can be summarized as follows:

- we proposed a kernelized maximal-figure-of-merit classifier using a subspace distance minimization technique for image concept modeling problems.
  - we provided mathematical formulation that proved that a subspace distance could

be minimized by solving a spectral decomposition problem of a kernel matrix, known as the Nyström extension.

  - we presented efficient procedures to solve the Nyström extension problem using rank-1 update and rank-1 downdate of a lower triangular matrix.
  - we formulated an algorithm to learn a nonlinear discriminant function for which a preferred performance metric could be directly optimized.

- we developed a novel semi-supervised incremental learning framework that utilized an expected error reduction. We then applied the framework for various image concept modeling problems.

  - we introduced an expected error reduction for an unlabeled sample with which the contribution of the sample to reducing classification error was measured.
  - we provided a systematic way to take advantage of the expected error reduction and the confidence scores to recommend unlabeled samples for updating parameters using an ensemble of classifiers.

- we presented a multi-view semi-supervised learning framework by proposing an agreement function that measured the level of the significance of the agreement on the classification prediction among classifiers trained on each view separately.

  - we presented an algorithm to compute the agreement function and provided a closed-form solution for a kernel function that unified multiple feature spaces into a single reproducing kernel Hilbert space with the agreement function.
  - we provided experimental evidences highlighting the effectiveness of the use of an agreement function on an image spam detection problem.

- we developed a semi-supervised incremental learning framework that integrated all the techniques presented in this dissertation, namely a discriminative semi-supervised incremental learning framework with a multi-view perspective.

– we demonstrated the robustness and the effectiveness of the developed framework with a set of experiments on an image spam detection task.

7.2 Avenues for future work

Nevertheless, there is still much room for improvement. One immediate future research venue is to investigate richer combinations of several discriminative learning algorithms for ensemble classifiers. For example, a combination of a structural SVM and a conditional random field might light up a direction to handle structural outputs. On the other hand, combinations between discriminative learning algorithms and generative learning methods, such as Gaussian Mixture Models, can reveal other interesting properties of the proposed semi-supervised incremental learning framework.

Additionally, one can also investigate various fusion methods to deal with multiple features in the future. For example, instead of combining features together prior to training classifiers, each classifier in an ensemble can be trained only with a single feature. Then, the outputs of individual classifiers are used in unlabeled sample selection procedures. However, there is one research issue that needs to be addressed for this approach; since classifiers in the ensemble will now have quite different levels of reliability, the formula to evaluate expected error reduction should be re-developed. Another interesting research direction is to reduce computational load during incremental learning procedures. Although the main focus of this dissertation is the development of a dependable unlabeled sample selection algorithm for incremental learning, it is also very important to design an efficient algorithm to update parameters at every iteration because the number of available data has been increasing exponentially these days.

In conclusion, we think that researchers have not fully taken advantage of the benefit of semi-supervised incremental learning toward image concept modeling problems due to its limited applicability. We hope that the work in this dissertation contributes to further development of semi-supervised incremental learning frameworks and more active application of the learning frameworks into image processing fields.

APPENDIX

Proof of Lemma 2

Proof. Suppose there is Ψ, a feature map from a sample space $\mathcal{X}$ to a Hilbert space $\mathcal{H}$ and a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ given by $\langle \Psi(x), \Psi(x) \rangle_{\mathcal{H}}$, an inner product on $\mathcal{H}$. Suppose further there are n samples denoted as $x_1, \ldots, x_n$ and two closed subspaces of $\mathcal{H}$, $\mathcal{U}$ and $\mathcal{V}$, spanned by $\{\Psi(x_i) : 1 \leq i \leq n\}$ and $\{\Psi(x_j) : 1 \leq j \leq q\}$, respectively, Here, it is assumed that $q < n$. Then, we can construct a linear operator for $\mathcal{U}$, $T_{\mathcal{U}} : \mathbb{R}^n \to \mathcal{U}$ given by $T(\boldsymbol{y}) = \sum_{i=1}^{n} y_i \Psi(x_i)$ for all $\boldsymbol{y} = [y_1, \ldots, y_n] \in \mathbb{R}^n$ and a linear operator for $\mathcal{V}$, $T_{\mathcal{V}} : \mathbb{R}^q \to \mathcal{V}$ given by $T_{\mathcal{V}}(\boldsymbol{y}) = \{\sum_{j=1}^{q} y_j \Psi(x_j) | \boldsymbol{y} = [y_1, \ldots, y_q] \in \mathbb{R}^q\}$. Furthermore, we can construct a self-adjoint operator $T_{\mathcal{U}}^* T_{\mathcal{U}} : \mathbb{R}^n \to \mathbb{R}^n$ for $\mathcal{U}$, where $T_{\mathcal{U}}^*$ is an adjoint operator of $T_{\mathcal{U}}$ given by $T_{\mathcal{U}}^*(\Psi(x_j)) = [k(x_1, x_j), \ldots, k(x_n, x_j)] \in \mathbb{R}^n$. Then, since the pre-image and image of $T_{\mathcal{U}}^* T_{\mathcal{U}}$ are both in finite dimensional spaces, it can be represented with a symmetric matrix K and decomposed using an unique orthonormal matrix $\Sigma \in \mathbb{R}^{n \times n}$ such that $\Sigma^T K \Sigma = \Lambda$, where $\Lambda \in \mathbb{R}^{n \times n}$ is a diagonal matrix. On the other hand, since $T_{\mathcal{U}}$ is a bounded linear operator, there exists a partial isometry U such that the image of U is the orthogonal complement of the kernel space of $T_{\mathcal{U}}$, which satisfies the following equality:

$$U = T_{\mathcal{U}}(\Sigma)\Lambda^{-\frac{1}{2}}, \tag{92}$$

where $T_{\mathcal{U}}(\Sigma)$ represents $[T_{\mathcal{U}}(\sigma_1), \ldots, T_{\mathcal{U}}(\sigma_r)]$ and r is the rank of the matrix K. Then, Σ can be represented in terms of U as

$$\Sigma = T_{\mathcal{U}}^*(U)\Lambda^{-\frac{1}{2}}, \tag{93}$$

and U can be represented as

$$U = T_{\mathcal{U}} T_{\mathcal{U}}^*(U)\Lambda^{-1}. \tag{94}$$

Similarly, For $T_{\mathcal{V}}$, we can derive the following equalities.

$$V = T_{\mathcal{V}}(\Sigma_s)\Lambda_s^{-\frac{1}{2}}, \tag{95}$$

and

$$\Sigma_s = T^*_{\mathcal{V}}(V)\Lambda_s^{-\frac{1}{2}}, \tag{96}$$

where V is a partial isometry of $T_{\mathcal{V}}$ and $\Sigma_s^T K_s \Sigma_s = \Lambda_s$. Here, K_s is a matrix representation of $T^*_{\mathcal{V}}T_{\mathcal{V}} : \mathbb{R}^q \to \mathbb{R}^q$, and $\Sigma_s \in \mathbb{R}^{q\times q}$ is an orthonormal matrix and $\Lambda_s \in \mathbb{R}^{q\times q}$ is a diagonal matrix. Now, consider $\tilde{\Sigma}_s$, an extension of Σ_s defined by

$$\tilde{\Sigma}_s = T^*_{\mathcal{U}}(V)\Lambda_s^{-\frac{1}{2}}. \tag{97}$$

Moreover, consider the i^{th} column of Σ denoted as σ_i, the i^{th} orthonormal basis of U, u_i, and the i^{th} diagonal entry of Λ, λ_i. Then,

$$\begin{aligned} \sigma_i^T \tilde{\Sigma}_s \tilde{\Sigma}_s^T \sigma_i &= \lambda_i^{-\frac{1}{2}} \langle u_i, T_{\mathcal{U}} T^*_{\mathcal{U}}(V)\rangle^T_{\mathcal{H}} \Lambda_s^{-1} \langle V, T_{\mathcal{U}} T^*_{\mathcal{U}}(u_i)\rangle_{\mathcal{H}} \lambda_i^{-\frac{1}{2}} & (98)\\ &= \lambda_i^{\frac{1}{2}} \langle u_i, V\rangle^T_{\mathcal{H}} \Lambda_s^{-1} \langle u_i, V\rangle_{\mathcal{H}} \lambda_i^{\frac{1}{2}}, & (99) \end{aligned}$$

where $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ is a inner product on $\mathcal{H}$. From Eq. (98) to Eq. (99), the equality in Eq. (94) is used. In Eq. (99), since V is a partial isometry of $T_{\mathcal{V}} : \mathbb{R}^q \to \mathcal{V}$, the inner product Eq. (99) is equal to the sum of inner products between the i^{th} orthonormal basis in $\mathcal{U}$ and the orthonormal basis vectors v_jS for $\mathcal{V}$ for $1 \le j \le q$. Whence, it is only remaining to simplify the left-hand side of Eq. (98) to complete the proof. In fact, By rearranging Eq. (99), we have

$$\langle u_i, V\rangle^T_{\mathcal{H}} \langle u_i, V\rangle_{\mathcal{H}} = \frac{1}{\lambda_i} \sigma_i^T \tilde{\Sigma}_s \Lambda_s \tilde{\Sigma}_s^T \sigma_i. \tag{100}$$

Note that by plugging Eq. (95) into Eq. (97),

$$\tilde{\Sigma}_s = T^*_{\mathcal{U}} T_{\mathcal{V}}(\Sigma_s)\Lambda_s^{-1}. \tag{101}$$

In Eq. (101), $T^*_{\mathcal{U}}T_{\mathcal{V}} : \mathbb{R}^q \to \mathbb{R}^n$ also has a matrix representation because the pre-image and image of $T^*_{\mathcal{U}}T_{\mathcal{V}}$ are all in finite dimensional spaces. Moreover, the matrix representation is defined as

$$T^*_{\mathcal{U}}T_{\mathcal{V}} = \begin{bmatrix} K_s \\ A^T \end{bmatrix}, \tag{102}$$

where A is a matrix representation of $k(x_i, x_j)$ for $q+1 \leq i \leq n$ and $1 \leq j \leq q$. By Eq. (102) and Eq. (101),

$$\tilde{\Sigma}_s = \begin{bmatrix} \Sigma_s \\ A^T \Sigma_s \Lambda_s^{-1} \end{bmatrix}, \tag{103}$$

and

$$\tilde{\Sigma}_s \Lambda_s \tilde{\Sigma}_s^T = \begin{bmatrix} K_s & A \\ A^T & A^T K_s^{-1} A \end{bmatrix}, \tag{104}$$

which completes the proof. □

A rank-1 update and downdate algorithm for the Nyström Extension

Suppose $K_s^{(t)}$ and $K_s^{(t+1)}$ are kernel matrices corresponding to $\mathcal{I}_s^{(t)}$ and $\mathcal{I}_s^{(t+1)}$, the subsets of $\mathcal{I}$ at time t and $t+1$, respectively. The swapping operation consists of two step procedures; shrinking and augmenting. Shrinking is an operation that an element in $\mathcal{I}_s^{(t)}$ is removed. On the other hand, augmenting is an operation in which an element in $\mathcal{I}_s^{c(t)}$ is inserted into $K_s^{(t)}$ to the same location where an element has been removed during shrinking. More formally, they can be written as follows:

(Shrinking)

$$K_s^{(t)} \quad \Rightarrow \quad K_s^{(t_{shrinked})} \tag{105}$$

$$\Leftrightarrow \begin{bmatrix} K_{11} & w & K_{21}^T \\ w^T & r & z^T \\ K_{21} & z & K_{22} \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} K_{11} & K_{21}^T \\ K_{21} & K_{22} \end{bmatrix} \tag{106}$$

(Augmenting)

$$K_s^{(t_{shrinked})} \quad \Rightarrow \quad K_s^{(t+1)} \tag{107}$$

$$\Leftrightarrow \begin{bmatrix} K_{11} & K_{21}^T \\ K_{21} & K_{22} \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} K_{11} & \tilde{w} & K_{21}^T \\ \tilde{w}^T & \tilde{r} & \tilde{z^T} \\ K_{21} & \tilde{z} & K_{22} \end{bmatrix}, \tag{108}$$

where K_{ij} for $i, j = \{1, 2\}$ are block matrices. Here, one can see that the three components corresponding to the selected element at time t, w, r, and z, are replaced with $\tilde{w}$, $\tilde{r}$, and $\tilde{z}$ through shrinking and augmenting.

Typically, the determinant of a symmetric matrix $K_s^{(t)}$ is computed by performing the Cholesky decomposition, which produces a lower triangular matrix $L_s^{(t)}$ such that $K_s^{(t)} = L_s^{(t)} L_s^{(t)T}$. The goal of utilizing the rank-1 update for shrinking operation is to find $L_s^{(t_{shrinked})}$ such that $K_s^{(t_{shrinkied})} = L_s^{(t_{shrinked})} L_s^{(t_{shrinked})T}$ without explicitly performing the Cholesky decomposition on $K_s^{(t_{shrinkied})}$.

To this end, from Eq. (106), one can set up the following two equations.

$$K_s^{(t)} = L_s^{(t)} L_s^{(t)T} \tag{109}$$

$$\Leftrightarrow \begin{bmatrix} K_{11} & w & K_{21}^T \\ w^T & r & z^T \\ K_{21} & z & K_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ l_w^T & l_r & 0 \\ L_{21} & l_z & L_{22} \end{bmatrix} \begin{bmatrix} L_{11}^T & l_w & L_{21}^T \\ 0 & l_r & l_z^T \\ 0 & 0 & L_{22}^T \end{bmatrix} \tag{110}$$

$$= \begin{bmatrix} L_{11}L_{11}^T & L_{11}l_w & L_{11}L_{21}^T \\ l_w^T L_{11}^T & l_w^T l_w + l_r^2 & l_w^T L_{21}^T + l_r l_z^T \\ L_{21}L_{11}^T & L_{21}l_w + l_r l_z & L_{21}L_{21}^T + l_z l_z^T + L_{22}L_{22}^T \end{bmatrix}, \tag{111}$$

and

$$K_s^{(t_{shrinked})} = L_s^{(t_{shrinked})} L_s^{(t_{shrinked})T} \tag{112}$$

$$\Leftrightarrow \begin{bmatrix} K_{11} & K_{21}^T \\ K_{21} & K_{22} \end{bmatrix} = \begin{bmatrix} L_{11}L_{11}^T & L_{11}L_{21}^T \\ L_{21}L_{11}^T & L_{21}L_{21}^T + l_z l_z^T + L_{22}L_{22}^T \end{bmatrix} \tag{113}$$

$$= \begin{bmatrix} L_{11} & 0 \\ L_{21} & L'_{22} \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ 0 & L'^T_{22} \end{bmatrix} \tag{114}$$

$$= \begin{bmatrix} L_{11}L_{11}^T & L_{11}L_{21}^T \\ L_{21}L_{11}^T & L_{21}L_{21}^T + L'_{22}L'^T_{22} \end{bmatrix}, \tag{115}$$

where L_{ij} and L'_{ij} for $i, j = \{1, 2\}$ are lower triangular matrices. Note that, l_w and l_z are column vectors and l_r is a scalar corresponding to w, z, and r, respectively. Comparing Eq. (113) and Eq. (115), one can see the following equality has to satisfy:

$$L'_{22}L'^T_{22} = l_z l_z^T + L_{22}L_{22}^T, \tag{116}$$

which is exactly the same as a rank-1 update of L_{22} with a vector l_z [44].

To derive the augmenting operation using a rank-1 downdate, one can set up similar equations. This time, one wants to find $L_s^{(t+1)}$ such that $K_s^{(t+1)} = L_s^{(t+1)} L_s^{(t+1)T}$ using a rank-1 downdate on $L_s^{(t_{shrinked})}$ with a certain vector. Let us consider the following equations:

$$K_s^{(t_{shrinked})} = L_s^{(t_{shrinked})} L_s^{(t_{shrinked})T} \tag{117}$$

$$\Leftrightarrow \begin{bmatrix} K_{11} & K_{21}^T \\ K_{21} & K_{22} \end{bmatrix} = \begin{bmatrix} L_{11}L_{11} & L_{11}L_{21}^T \\ L_{21}L_{11} & L_{21}L_{21}^T + L'_{22}L'^T_{22} \end{bmatrix}, \tag{118}$$

and

$$K_s^{(t+1)} = L_s^{(t+1)} L_s^{(t+1)T} \tag{119}$$

$$\Leftrightarrow \begin{bmatrix} K_{11} & \tilde{w} & K_{21}^T \\ \tilde{w}^T & \tilde{r} & \tilde{z}^T \\ K_{21} & \tilde{z} & K_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ \tilde{l}_w^T & \tilde{l}_r & 0 \\ L_{21} & \tilde{l}_z & \tilde{L}_{22} \end{bmatrix} \begin{bmatrix} L_{11}^T & \tilde{l}_w & L_{21}^T \\ 0 & \tilde{l}_r & \tilde{l}_z^T \\ 0 & 0 & \tilde{L}_{22}^T \end{bmatrix} \tag{120}$$

$$= \begin{bmatrix} L_{11}L_{11}^T & L_{11}\tilde{l}_w & L_{11}L_{21}^T \\ \tilde{l}_w^T L_{11}^T & \tilde{l}_w^T\tilde{l}_w + \tilde{l}_r^2 & \tilde{l}_w^T L_{21}^T + \tilde{l}_r\tilde{l}_z^T \\ L_{21}L_{11}^T & L_{21}\tilde{l}_w + \tilde{l}_r\tilde{l}_z & L_{21}L_{21}^T + \tilde{l}_z\tilde{l}_z^T + \tilde{L}_{22}\tilde{L}_{22}^T \end{bmatrix}. \tag{121}$$

Now, for the augmenting operation, one needs to solve for $\tilde{l}_w$, $\tilde{l}_z$, $\tilde{l}_r$, and $\tilde{L}_{22}$. For $\tilde{l}_w$, a linear system of equations, $\tilde{w} = L_{11}\tilde{l}_w$ can be used, which can be easily solved because L_{11} is a lower triangular matrix. For $\tilde{l}_r$, we only need to compute $\tilde{l}_r = \sqrt{\tilde{r} - \tilde{l}_w^T\tilde{l}_w}$, which is guaranteed to exist, since K_s is positive semi-definite. $\tilde{l}_z$ can be obtained by simple algebraic operations on $\tilde{z} = L_{21}\tilde{l}_w + \tilde{l}_r\tilde{l}_z$. For $\tilde{L}_{22}$, the following equation has to satisfy:

$$\tilde{L}_{22}\tilde{L}_{22}^T = L'_{22}L'^T_{22} - \tilde{l}_z\tilde{l}_z^T, \tag{122}$$

which is exactly the same as a rank-1 downdate of L'_{22} with $\tilde{l}_z$. Whence completing the presentation of the algorithm.

Proof of Theorem 3

Proof. The first part of the theorem is an immediate corollary of Theorem 2.2 in [92] by

redefining the inner product in the theorem as

$$\begin{aligned}\langle f, g\rangle_{\mathcal{H}_\mathcal{K}} &= \lambda_1\langle f^{(1)}, g^{(1)}\rangle_{\mathcal{H}_K^{(1)}} + \lambda_2\langle f^{(2)}, g^{(2)}\rangle_{\mathcal{H}_K^{(2)}} \\ &\quad +\mu\sum_{i\in\mathcal{U}}\xi_i(f^{(1)}(x_{u_i}) - f^{(2)}(x_{u_i}))(g^{(1)}(x_{u_i}) - g^{(2)}(x_{u_i})). \qquad (123)\end{aligned}$$

To prove k is a valid kernel function, let us first assume ξ_i is positive for all $x_{u_i} \in \mathcal{U}$. This is perfectly valid because $\xi_i \geq 0$ by the definition of ξ_i. Moreover we can drop x_{u_i} such that $\xi_i = 0$ from the co-regularization term. So Ξ^{-1} exists. Now, let us express $k(x,\cdot)$ as $k(x,\cdot) = h^{(1)}(x,\cdot) + h^{(2)}(x,\cdot)$, where $h^{(1)}(x,\cdot) \in span\{k^{(1)}(x,\cdot),$ and $k^{(1)}(x_{u_i},\cdot)$ for $\forall i \in \mathcal{U}\}$ and $h^{(2)}(x,\cdot) \in span\{k^{(2)}(x,\cdot),$ and $k^{(2)}(x_{u_i},\cdot)$ for $\forall i \in \mathcal{U}\}$. This is also possible because $k(x,\cdot) \in \mathcal{H}_K$, and by the definition of $\mathcal{H}_K$, there exists some $h^1(x,\cdot) \in \mathcal{H}_K^{(1)}$ and $h^2(x,\cdot) \in \mathcal{H}_K^{(2)}$ such that $k(x,\cdot) = h^{(1)}(x,\cdot) + h^{(2)}(x,\cdot)$. Now, to show k is a valid kernel function, one only need to show the inner product $\langle f, k(x,\cdot)\rangle_{\mathcal{H}_K}$ is equal to $f(x)$, say, $\langle f, k(x,\cdot)\rangle_{\mathcal{H}_K} = f^{(1)}(x) + f^{(2)}(x) = f(x)$. To show this equality, let us consider $h^{(1)}(x,\cdot)$ and $h^{(2)}(x,\cdot)$ defined by

$$h^{(1)}(x,\cdot) = \lambda_1^{-1}\{k^{(1)}(x,\cdot) - \mu\mathbf{d}_{\mathrm{x}}^T\tilde{H}k^{(1)}(U,\cdot)\} \qquad (124)$$

and

$$h^{(1)}(x,\cdot) = \lambda_2^{-1}\{k^{(2)}(x,\cdot) + \mu\mathbf{d}_{\mathrm{x}}^T\tilde{H}k^{(2)}(U,\cdot)\}, \qquad (125)$$

where $k^{(1)}(U,\cdot)$ and $k^{(2)}(U,\cdot)$ represent column vectors consisting of $k^{(1)}(x_{u_i},\cdot)$ and $k^{(2)}(x_{u_i},\cdot)$, for $i \in \mathcal{U}$, respectively, and $\mathbf{d}_{\mathrm{x}}^T$ and $\tilde{H}$ are defined as in Theorem 3. It is clear that $k(x,\cdot)$ can still be expressed as $k(x,\cdot) = h^{(1)}(x,\cdot) + h^{(2)}(x,\cdot)$. Furthermore, it can be shown that

$$[h^{(1)}(x,U) - h^{(2)}(x,U)]^T\Xi = \mathbf{d}_{\mathrm{x}}^T\tilde{H}, \qquad (126)$$

where $h^{(1)}(x,U)$ and $h^{(2)}(x,U)$ are also column vectors consisting of $h^{(1)}(x,x_i)$ and $h^{(2)}(x,x_i)$,

for $i \in \mathcal{U}$, where Ξ is defined as in Theorem 3. Then,

$$\begin{aligned}
\langle f, k(x,\cdot)\rangle_{\mathcal{H}_K} &= \lambda_1 \left\langle f^{(1)}, h^{(1)}(x,\cdot)\right\rangle_{\mathcal{H}_K^{(1)}} + \lambda_2 \left\langle f^{(2)}, h^{(2)}(x,\cdot)\right\rangle_{\mathcal{H}_K^{(2)}} \\
&\quad +\mu[f^{(1)}(U) - f^{(2)}(U)]^T \Xi [h^{(1)}(x,U) - h^{(2)}(x,U)] \qquad (127) \\
&= f^{(1)}(x) - \mu \mathbf{d}_{\mathrm{x}}^T \tilde{H} f^{(1)}(U) + f^{(2)}(x) + \mu \mathbf{d}_{\mathrm{x}}^T \tilde{H} f^{(2)}(U) \\
&\quad +\mu[f^{(1)}(U) - f^{(2)}(U)]^T \tilde{H} \mathbf{d}_{\mathrm{x}} \qquad (128) \\
&= f^{(1)}(x) + f^{(2)}(x) \qquad (129) \\
&= f(x), \qquad (130)
\end{aligned}$$

where $f^{(1)}(U)$ and $f^{(2)}(U)$ are column vectors consisting of $f^{(1)}(x_{u_i})$ and $f^{(2)}(x_{u_i})$, for $i \in \mathcal{U}$. This completes the proof. □

REFERENCES

[1] Abe, N. and Mamitsuka, H., "Query learning strategies using boosting and bagging," in *Proc. of ICML*, 1998.

[2] Aronszajn, N., "Theory of reproducing kernels," *Trans. of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.

[3] Ashby, F. G. and Maddox, W. T., "Human category learning," *Annual Review of Psychology*, vol. 56, pp. 149–178, 2005.

[4] Balcan, M.-F., Blum, A., and Yang, K., "Co-training and expansion: Towards bridging theory and practice," in *Advances in Neural Information Processing Systems*, 2005.

[5] Bartlett, P. L. and Mendelson, S., "Rademacher and gaussian complexities: risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, 2003.

[6] Belabbas, M.-A. and Wolfe, P. J., "Spectral methods in machine learning and new strategies for very large datasets," in *Proc. of the National Academy of Sciences*, 2009.

[7] Belkin, M., Niyogi, P., Sindhwani, V., and Bartlett, P., "Manifold regularization: A geometric framework for learning from examples," *Journal of Machine Learning Research*, 2004.

[8] Bellegarda, J.-R., "Exploiting latent semantic information in statistical language modeling," in *Proc. of the IEEE*, 2000.

[9] Berg, B. A., *Markov Chain Monte carlo simulations and their statistical analysis*. World Scientific Publishing Company, 2004.

[10] Bicker, S. and Scheffer, T., "Estimation of mixture models using co-em," in *Proc. of ICML workshop on Learning with Multiple Views*, 2005.

[11] Bishop, C. M., *Pattern Recognition and Machine Learning*. Springer New York Inc., 2006.

[12] Blum, A. and Chawla, S., "Learning from labeled and unlabeled data using graph mincuts," in *Proc. of ICML*, 2001.

[13] Blum, A. and Mitchell, T., "Combining labeled and unlabeled data with co-training," in *Proc. of COLT*, 1998.

[14] Bradley, C. L., "Comparing supervised and unsupervised category learning," *Psychonomic Bulletin and Review*, vol. 9, pp. 829–835, 2002.

[15] Brefeld, U., Gärtner, T., Scheffer, T., and Wrobel, S., "Efficient co-regularised least squares regression," in *Proc. of ICML*, 2006.

[16] Brefeld, U. and Scheffer, T., "Auc maximizing support vector learning," in *Proc. of ICML workshop on ROC Analysis in Machine Learning*, 2005.

[17] Byun, B. and Lee, C.-H., "An incremental learning framework combining sample confidence and discrimination with an application to automatic image annotation," in *Proc. of ICIP*, 2009.

[18] Byun, B. and Lee, C.-H., "A kernelized maximal-figure-or-merit learning approach based on subspace distance minimization," in *Proc. of ICASSP*, 2011.

[19] Byun, B., Lee, C.-H., Webb, S., Irani, D., and Pu, C., "An anti-spam filter combination framework for text-and-image emails," in *Proc. of CEAS*, 2009.

[20] Byun, B., Lee, C.-H., Webb, S., and Pu, C., "A discriminative classifier learning approach to image modeling and spam image identification," in *Proc. of CEAS*, 2007.

[21] Byun, B., Ma, C., and Lee, C.-H., "An experimental study on discriminative concept classifier combination for trecvid high-level feature extraction," in *Proc. of ICIP*, 2008.

[22] C. Mario Christoudias, Raquel Urtasun, T. D., "Multi-view learning in the presence of view disagreement," in *Proc. of UAI*, 2008.

[23] Carreras, X. and Mrquez, L., "Boosting trees for anti-spam email filtering," in *Proc. of RANLP*, 2001.

[24] Chapelle, O., "A continuation method for semi-supervised svms," in *Proc. of ICML*, 2006.

[25] Chapelle, O., Schölkopf, B., and Zien, A., eds., *Semi-Supervised Learning*. MIT Press, 2006.

[26] Coifman, R. R. and Lafon, S., "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, pp. 5–30, 2006.

[27] Corduneanu, A., *The information regularization framework for semi-supervised learning*. PhD thesis, Massachusetts Institute of Technology, 2006.

[28] Dasgupta, S., Littman, M. L., and McAllester, D., "Pac generalization bounds for co-training," in *Advances in Neural Information Processing Systems*, 2001.

[29] de Sa, V. R., "Spectral clustering with two views," in *Proc. of ICML workshop on Learning with Multiple Views*, 2005.

[30] Drucker, H., W. D. and Vapnik, V. N., "Support vector machines for spam categorization," *IEEE Trans. on Neural Networks*, vol. 10, 1999.

[31] Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T., "Efficient projections onto the l1-ball for learning in high dimensions," in *Proc. of ICML*, pp. 272–279, 2008.

[32] Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2 ed., 2001.

[33] Engel, Y., Mannor, S., and Meir, R., "The kernel recursive least squares algorithm," *IEEE Trans. on Signal Processing*, vol. 52, pp. 2275–2285, 2003.

[34] Eugene, D. M., Charniak, E., and Johnson, M., "Effective self-training for parsing," in *Proc. of N. American ACL (NAACL)*, 2006.

[35] Fan, X., Guo, Z., and Ma, H., "An improved em-based semi-supervised learning method," in *Proc. of IJCBS*, 2009.

[36] Farquhar, J., Hardoon, D., Meng, H., Shawe-Taylor, J., and Szedmak, S., "Two view learning: Svm-2k, theory and practice," in *Advances in Neural Information Processing Systems*, 2005.

[37] Field, D. J., "Relations between the statistics of natural images and the response properties of cortical cells," *Journal of the Optical Society of America A*, vol. 4, pp. 2379–2394, 1987.

[38] Fisher, R. A., "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, 1936.

[39] Fowlkes, C., Belongie, S., Chung, F., and Malik, J., "Spectral grouping using the nystrom method," *IEEE Trans. on Pattern Analysis and Machine Intellegence*, vol. 26, 2004.

[40] Freeman, W. and Adelson, E., "The design and use of steerable filters," *IEEE Trans. on Pattern Analysis and Machine Intellegence*, vol. 13, pp. 891–906, 1991.

[41] Freund, Y., Seung, H. S., Shamir, E., and Tishby, N., "Selective sampling using the query by committee algorithm," *Mach. Learn.*, vol. 28, 1997.

[42] Gao, S., Wang, D.-H., and Lee, C.-H., "Automatic image annotation through multi-topic text categorization," in *Proc. of ICASSP*, 2006.

[43] Gao, S., Wu, W., and Lee, C.-H., "A mfom learning approach to robust multiclass multi-label text categorization," in *Proc. of ICML*, 2004.

[44] Gill, P. E., Golub, G. H., Murray, W. A., and Saunders, M. A., "Methods for modifying matrix factorizations," technical report, Stanford University, 1972.

[45] Gonzalez, R. C. and Woods, R. E., *Digital Image Processing 2ed.* Prentice Hall Press, 2002.

[46] Gr, Y. and Bengio, Y., "Semi-supervised learning by entropy minimization," in *Advances in Neural Information Processing Systems*, 2004.

[47] Guattery, S. and Miller, G. L., "Graph embeddings and laplacian eigenvalues," *SIAM J. on Matrix Analysis and Applications*, vol. 21, p. 2000, 2000.

[48] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning.* Springer Series in Statistics, Springer New York Inc., 2001.

[49] Hoi, S. C. H., Jin, R., Zhu, J., and Lyu, M. R., "Semisupervised svm batch mode active learning with applications to image retrieval," *ACM Trans. Inf. Syst.*, vol. 27, 2009.

[50] Hwa, R., "Sample selection for statistical parsing," *Comput. Linguist.*, vol. 30, 2004.

[51] Iyengar, G. and Nock, H. J., "Discriminative model fusion for semantic concept detection and annotation in video," in *Proc. of ACMMM*, 2003.

[52] Jerzy Neyman, E. P., "On the problem of the most efficient tests of statistical hypotheses," *Philosophical Transactions of the Royal Society of London*, 1933.

[53] Jiang, H., "Confident measure for speech recognition: A survey," in *Speech communication*, 2005.

[54] Joachims, T., "Transductive learning via spectral graph partitioning," in *Proc. of ICML*, 2003.

[55] Joachims, T., "A support vector method for multivariate performance measures," in *Proc. of ICML*, 2005.

[56] Joachims, T., "Transductive inference for text classification using support vector machines," in *Proc. of ICML*, 1999.

[57] Juang, B.-H., Chou, W., and Lee, C.-H., "Minimum classification error rate methods for speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 5, no. 3, 1997.

[58] Kearns, M. J. and Vazirani, U. V., *An Introduction to Computational Learning Theory*. MIT Press, 1994.

[59] Kindermann, R., *Markov random fields and their applications*. American Mathematical Society, 1980.

[60] Kovesi, P. D., "Image feature from phase congruency," *Videre: Journal of Computer Vision Research*, vol. 1, no. 3, 1999.

[61] Lafferty, J., "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. of ICML*, 2001.

[62] Leavitt, N., "Vendors fight spam's sudden rise," *IEEE Computer*, vol. 40, pp. 16–19, 2007.

[63] Leskes, B. and Torenvliet, L., "The value of agreement a new boosting algorithm," *J. Comput. Syst. Sci.*, vol. 74, 2008.

[64] Lewis, D. D. and Gale, W. A., "A sequential algorithm for training text classifiers," in *Proc. of ACM SIGIR*, 1994.

[65] Li, X., Wang, L., and Sung, E., "Multi-label svm active learning for image classification," in *Proc. of ICIP*, 2004.

[66] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *Int. Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[67] Muslea, I., Minton, S., and Knoblock, C. A., "Selective sampling with redundant views," in *Proc. of AAAI*, 2000.

[68] Nigam, K. and Ghani, R., "Analyzing the effectiveness and applicability of co-training," in *Proc. of CIKM*, 2000.

[69] Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T., "Text classification from labeled and unlabeled documents using em," *Mach. Learn.*, vol. 39, 2000.

[70] Olivier Chapelle, Jason Weston, B. S., "Cluster kernels for semi-supervised learning," in *Advances in Neural Information Processing Systems*, 2003.

[71] Parzen, E., "On estimation of a probability density function and mode," *Annals of Mathematical Statistics*, vol. 33, no. 3, 1962.

[72] Pinar Duygulu, Kobus Barnard, N. d. F. and Forsyth, D., "Data for object recognition as machine translation," *http://kobus.ca/research/data/*, 2002.

[73] Platt, J. C., "Fast training of support vector machines using sequential minimal optimization," in *Advances in kernel methods: support vector learning*, MIT Press, 1999.

[74] Platt, J. C., "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*, MIT Press, 1999.

[75] Quattoni, A., Carreras, X., Collins, M., and Darrell, T., "An efficient projection for l1,inf regularization," in *Proc. of ICML*, 2009.

[76] Quenot, G., Tseng, A., Safadi, B., and Ayache, S., "Trecvid 2010 collaborative annotation," *http://mrim.imag.fr/tvca/*, 2010.

[77] Rasmussen, C. E. and Williams, C. K. I., *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[78] Rosenberg, C., Hebert, M., and Schneiderman, H., "Semi-supervised self-training of object detection models," in *Proc. of IEEE Workshop on Applications of Computer Vision and Motion and Video Computing*, 2005.

[79] Rosenberg, D. S. and Bartlett, P. L., "The rademacher complexity of co-regularized kernel classes," *Proc. of AISTATS*, 2007.

[80] Roy, N. and McCallum, A., "Toward optimal active learning through sampling estimation of error reduction," in *Proc. of ICML*, 2001.

[81] Rui, Y., Huang, T. S., and Chang, S.-F., "Image retrieval: Current techniques, promising directions, and open issues," *Journal of Visual Communication and Image Retrieval*, 1999.

[82] Sahami, M., D. S. H. D. and Horvitz, E., "A bayesian approach to filtering junk email," in *Proc. of AAAI Workshop on Learning for Text Categorization*, 1998.

[83] Schmidt, M., Fung, G., and Rosales, R., "Fast optimization methods for l1 regularization: A comparative study and two new approaches," in *Proc. of ECML*, 2007.

[84] Schölkopf, B., Smola, A., and Müller, K.-R., "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, 1998.

[85] Seeger, M., "Learning with labeled and unlabeled data," technical report, The University of Edinburgh, 2001.

[86] Seeger, M., "Covariance kernels from bayesian generative models," in *Advances in Neural Information Processing Systems*, 2000.

[87] Selinger, A. and Nelson, R. C., "Minimally supervised acquisition of 3d recognition models from cluttered images," in *Proc. of CVPR*, 2001.

[88] Settles, B., "Active learning literature survey," Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[89] Settles, B. and Craven, M., "An analysis of active learning strategies for sequence labeling tasks," in *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 2008.

[90] Sindhwani, V., Niyogi, P., and Belkin, M., "A co-regularization approach to semi-supervised learning with multiple views," in *Proc. of ICML*, 2005.

[91] Sindhwani, V., Niyogi, P., and Belkin, M., "Beyond the point cloud: from transductive to semi-supervised learning," in *Proc. of ICML*, 2005.

[92] Sindhwani, V. and Rosenberg, D. S., "An rkhs for multi-view learning and manifold co-regularization," in *Proc. of ICML*, 2008.

[93] Steedman, M., Osborne, M., Sarkar, A., Clark, S., Hwa, R., Hockenmaier, J., Ruhlen, P., Baker, S., and Crim, J., "Bootstrapping statistical parsers from small datasets," in *Proc. of EACL*, 2003.

[94] Stricker, M. and Orengo, M., "Similarity of color images," in *Proc. of SPIE*, 1995.

[95] Swain, M. and Ballard, D., "Color indexing," *Int. Journal of Computer Vision*, vol. 1, pp. 11–32, 1991.

[96] Tibshirani, R., "Regression shrinkage and selection via the lasso," *J. R. Statist. Soc. B*, no. 1, 1996.

[97] Tong, S. and Chang, E., "Support vector machine active learning for image retrieval," in *Proc. of ACMMM*, 2001.

[98] Tong, S. and Koller, D., "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, 2000.

[99] Vapnik, V., *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[100] Vijayakumar, S. and Schaal, S., "Fast and efficient incremental learning for high-dimensional movement systems," in *Proc. of ICRA*, pp. 1894–1899, 2000.

[101] Wang, D.-H., Gao, S., Tian, Q., and Sung, W.-K., "Discriminative fusion approach for automatic image annotation," in *Proc. of MMSP*, 2005.

[102] Wang, L. and Shen, X., "On l1-norm multiclass support vector machines: Methodology and theory," *Journal of the American Statistical Association*, vol. 102, 2007.

[103] Wang, L., Wang, X., and Feng, J., "Subspace distance analysis with application to adaptive bayesian algorithm for face recognition," *Pattern Recogn.*, vol. 39, no. 3, 2006.

[104] Wang, S. B., Quattoni, A., Morency, L.-P., and Demirdjian, D., "Hidden conditional random fields for gesture recognition," in *Proc. of CVPR*, 2006.

[105] Wang, W. and hua Zhou, Z., "Analyzing co-training style algorithms," in *Proc. of ECML*, 2007.

[106] Williams, C. and Seeger, M., "Using the nystrom method to speed up kernel machines," in *Advances in Neural Information Processing Systems*, 2001.

[107] Wu, C. T., C. K. T. Z. Q. and Wu, Y. L., "Using visual features for anti-spam filtering," in *Proc. of ICIP*, 2005.

[108] Wu, Y., Tian, Q., and Huang, T. S., "Discriminant-em algorithm with application to image retrieval," in *Proc. of CVPR*, 2000.

[109] Wu, Y., Tian, Q., Huang, T. S., and Toyama, K., "Self-supervised learning for object recognition based on kernel discriminant-em algorithm," in *Proc. of ICCV*, 2001.

[110] Yarowsky, D., "Unsupervised word sense disambiguation rivaling supervised methods," in *Proc. of ACL*, 1995.

[111] Yu, S., Krishnapuram, B., Rosales, R., Steck, H., and Rao, R. B., "Bayesian co-training," in *Advances in Neural Information Processing Systems*, 2008.

[112] Zhang, R. and Rudnicky, A., "A new data selection principle for semi-supervised incremental learning," in *Proc. of ICPR*, 2006.

[113] Zhang, R. and Rudnicky, A. I., "A new data selection approach for semi-supervised acoustic modeling," in *Proc. of ICASSP*, 2006.

[114] Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Olkopf, B. S., "Learning with local and global consistency," in *Advances in Neural Information Processing Systems*, 2004.

[115] Zhou, D., Lawless, S., Min, J., and Wade, V., "A late fusion approach to cross-lingual document re-ranking," in *Proc. of ACM CIKM*, 2010.

[116] Zhu, J., Rosset, S., Hastie, T., and Tibshirani, R., "L1 norm support vector machines," in *Advances in Neural Information Processing Systems*, 2003.

[117] Zhu, X., "Semi-supervised learning literature survey," Technical Report 1530, Computer Sciences, U. of Wisconsin-Madison, 2005.

[118] Zhu, X., Lafferty, J., and Ghahramani, Z., "Combining active learning and semi-supervised learning using gaussian fields and harmonic functions," in *Proc. of ICML*, 2003.

[119] Zhu, X., Rogers, T., Qian, R., and Kalish, C., "Humans perform semi-supervised classification too," in *Proc. of AAAI*, 2007.